A USER'S MANUAL AND PROGRAM DESCRIPTION OF SEA PLOT: A THREE-DIMENSIONAL GRAPHICS PROGRAM DEPICTING A WAR-AT-SEA ENCOUNTER BETWEEN A SINGLE SHIP AND A SINGLE AIRCRAFT

Gray DeVeaux Hobby

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A USER'S MANUAL AND PROGRAM DESCRIPTION OF
SEA PLOT: A THREE-DIMENSIONAL GRAPHICS
PROGRAM DEPICTING A WAR-AT-SEA ENCOUNTER
BETWEEN A SINGLE SHIP AND A SINGLE AIRCRAFT

by

Gray DeVeaux Hobby

June 1981

Thesis Advisor: R. E. Ball

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A User's Manual and Program Description of SEA PLOT: A Three-Dimensional Program Depicting a War-at-Sea encounter between a Single Ship and a Single Aircraft		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; June 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gray DeVeaux Hobby		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1981
		13. NUMBER OF PAGES 90
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Three-Dimensional Computer Graphics
Conic Projection
Compass Hammer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SEA PLOT is a three-dimensional graphics program written for the Compass Hammer research effort. The program portrays a war-at-sea scenario in which a single aircraft attacks a single ship target located at the origin of an earth-fixed coordinate system. To plot the three-dimensional scenario on a two-dimensional plotting surface, a conic projection technique is used in which a cone of vision is generated

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601 |

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

based on the position of a viewer's eye. The position of the eye is automatically placed based upon the initial coordinate position of the attacking aircraft. The program is written using VERSATEC software available on the NPS IBM 3033 computer. However, the program has been structured in such a manner as to permit easy transition to a more sophisticated, picture oriented, graphics language such as PLOT 80 which would permit an interactive graphics employment of the program and possibly animation of the attacking scenario.

Approved for public release; distribution unlimited.

A User's Manual and Program Description of SEA PLOT: A Three-Dimensional Graphics Program Depicting a War-at-Sea Encounter between a Single Ship and a Single Aircraft

by

Gray DeVeaux Hobby
Lieutenant Commander, United States Navy
B.S.MA., Purdue University, 1972

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1981

ABSTRACT

SEA PLOT is a three-dimensional graphics program written for the Compass Hammer research effort. The program portrays a war-at-sea scenario in which a single aircraft attacks a single ship target located at the origin of an earth-fixed coordinate system. To plot the three-dimensional scenario on a two-dimensional plotting surface, a conic projection technique is used in which a cone of vision is generated based on the position of a viewer's eye. The position of the eye is automatically placed based upon the initial coordinate position of the attacking aircraft. The program is written using VERSATEC software available on the NPS IBM 3033 computer. However, the program has been structured in such a manner as to permit easy transition to a more sophisticated, picture oriented, graphics language such as PLOT 80 which would permit an interactive graphics employment of the program and possibly animation of the attacking scenario.

TABLE OF CONTENTS

I. INTRODUCTION

This thesis provides a program description and user's manual for the graphics package SEA PLOT designed for use in the Compass Hammer research effort. SEA PLOT is a three-dimensional graphics program which provides the user a perspective display of a war-at-sea scenario in which a single ship target is attacked by a single aircraft. The plot depicts (1) ship position and heading, (2) aircraft flight profile including coordinate position, roll, pitch and yaw, and (3) ship defensive missile launch envelope. All information is displayed in a three-dimensional format using a conic projection in which an optimal viewing position is automatically selected based upon the attacking aircraft's initial coordinate position. Because the plotting package is intended to provide a visual overview of the attacking scenario, it is not a reliable source for making measurements of ranges, bearings, etc. This results from the conic projection technique used which displays solid objects in three-space much as they would be viewed by the viewer's eye. Consequently, range marks and rings located in close proximity to the viewing position will show a wider dispersion than those at a more distant location. The conic projection itself is discussed in detail in Chapter II.

Plotted output from SEA PLOT consists of range rings from the center of the target, aircraft flight profile, target ship orientation, and missile launch envelope. Range rings are displayed at 5000-meter increments from the center of the target located at the plot center. Maximum range is calculated by examination of the attacking aircraft's flight profile such that maximum plot range corresponds to the maximum separation between target and attacker. The target ship is shown in planform only, but is oriented along a user input heading. Three-dimensional display of the target ship, which is possible with only minor program changes, resulted in a cluttered plot output and consequently was omitted. Display of a defensive missile launch envelope consists of a cylinder comprised of unconnected dots corresponding to maximum missile slant range. The cylinder is truncated at a height of twice the maximum aircraft flight profile altitude. Aircraft ground track is depicted via a series of dark dots plotted on the surface corresponding to aircraft position over ground. The air track is shown in perspective view as a bold line tracking the vehicle movement through the air. Along the air track, an aircraft body is positioned and drawn corresponding to a minimum of 2000 meters of ground travel. This body plot displays aircraft maneuvers in roll, pitch, and yaw. Aircraft altitude perspective is enhanced via vertical dashed lines connecting ground and air tracks.

SEA PLOT also contains a variety of input scaling parameters which permit the user to focus on the close-in encounter between the target and aircraft, or to expand the view and observe the entire profile from initial to final aircraft positions. This is accomplished by entry of the scale parameter, one of the program data input elements. For example, SCALE=1 causes a wide angle display of the scenario. SCALE=2 in essence magnifies the plot size by a factor of two, focusing on the area from the origin out to one half the original range. Figures 1-1, 1-2, and 1-3 contain examples of plot output using SCALE=1, 2 and 3, respectively. Chapter III discusses format and the required program inputs for SEA PLOT.

Because SEA PLOT output is to be included in the Compass Hammer report, hard copy, printed output was required, as opposed to an interactive graphic output onto a CRT terminal. Consequently, the plotting package was written using VERSATEC software available on the NPS computer. Because this software system is oriented primarily toward display of data in graphic or tabular form, and not toward the drawing of pictures, the programming algorithms were more complicated than would be required for an interactive graphics software package, such as PLOT 80 used on many Tektronix machines. However, because the program does have many interactive graphics applications, such as an iterative design of an aircraft mission profile, it has

been structured as much as possible to permit a relatively simple conversion to other software systems. To accomplish this, the graphics instructions are written in internal subroutines, each of which contains the commands peculiar to the particular software system employed. For example, to draw a line connecting an array of coordinate points, SEA PLOT calls an internal subroutine DRAW. Resident in DRAW are the VERSATEC commands necessary to draw the line. To convert to another graphics system, one would merely have to adjust the instructions in DRAW to facilitate the software conversion rather than search through the entire program locating specialized line drawing instructions.

Since SEA PLOT was written in as general a form as possible, its applications are not limited to the Compass Hammer project. With generality in mind, it was prepared as a stand-alone program and was not incorporated into one of the Compass Hammer computing routines. SEA PLOT, in essence, can be utilized by any user desiring a three-dimensional attack scenario plot. It need not be constrained to the display of attacking aircraft encounters, but could be used from the standpoint of a ship defending itself from a single aircraft, evaluation of defensive missile design, or evaluation of weapon scenario development. Another possible use of SEA PLOT would be in conjunction with the P001 Input Program (PIP), which currently only provides two-dimensional plotted output.

Figure 1-1: SEA PLOT Output with SCALE=1

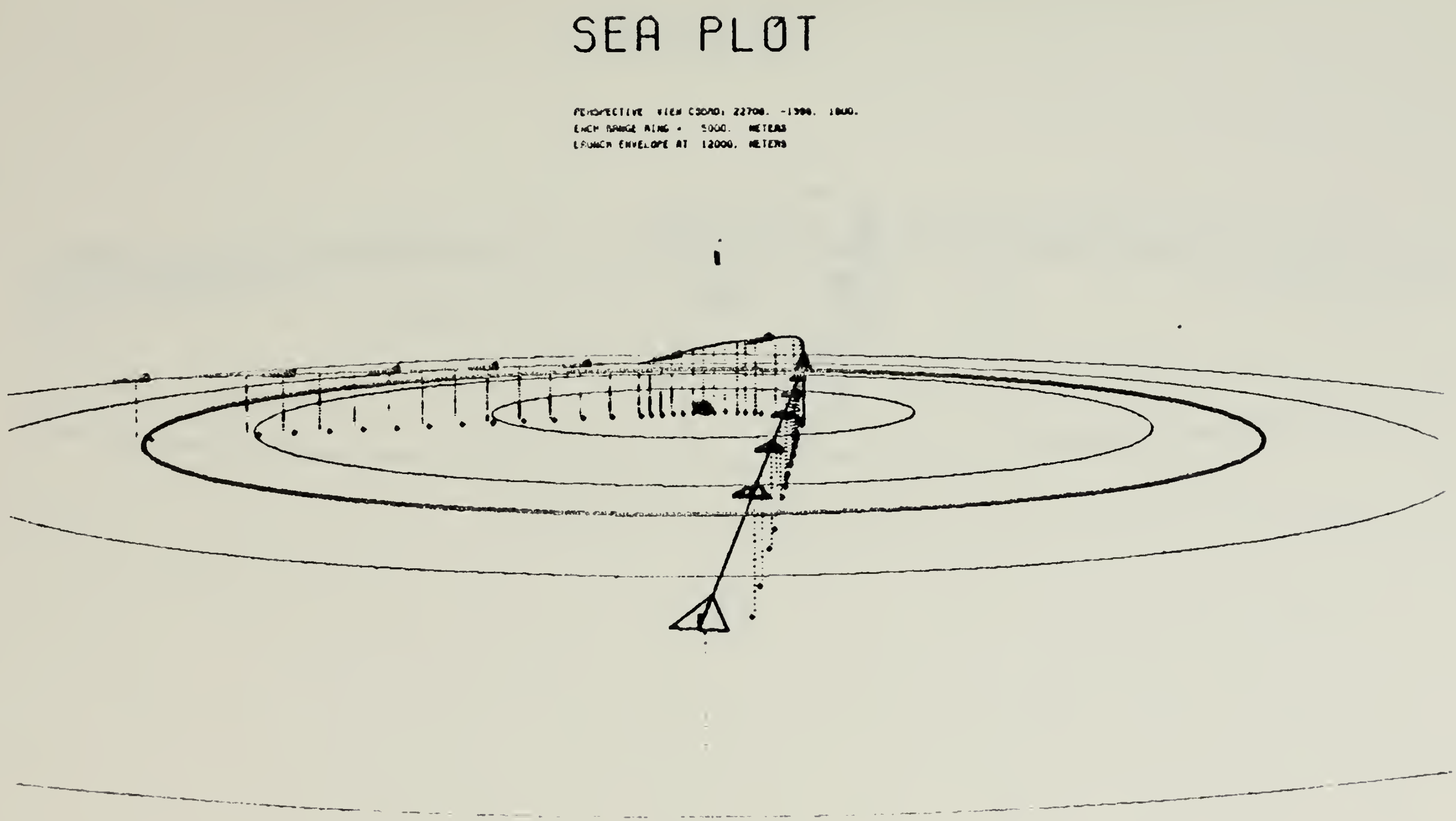

Figure 1-2: SEA PLOT Output with SCALE=2

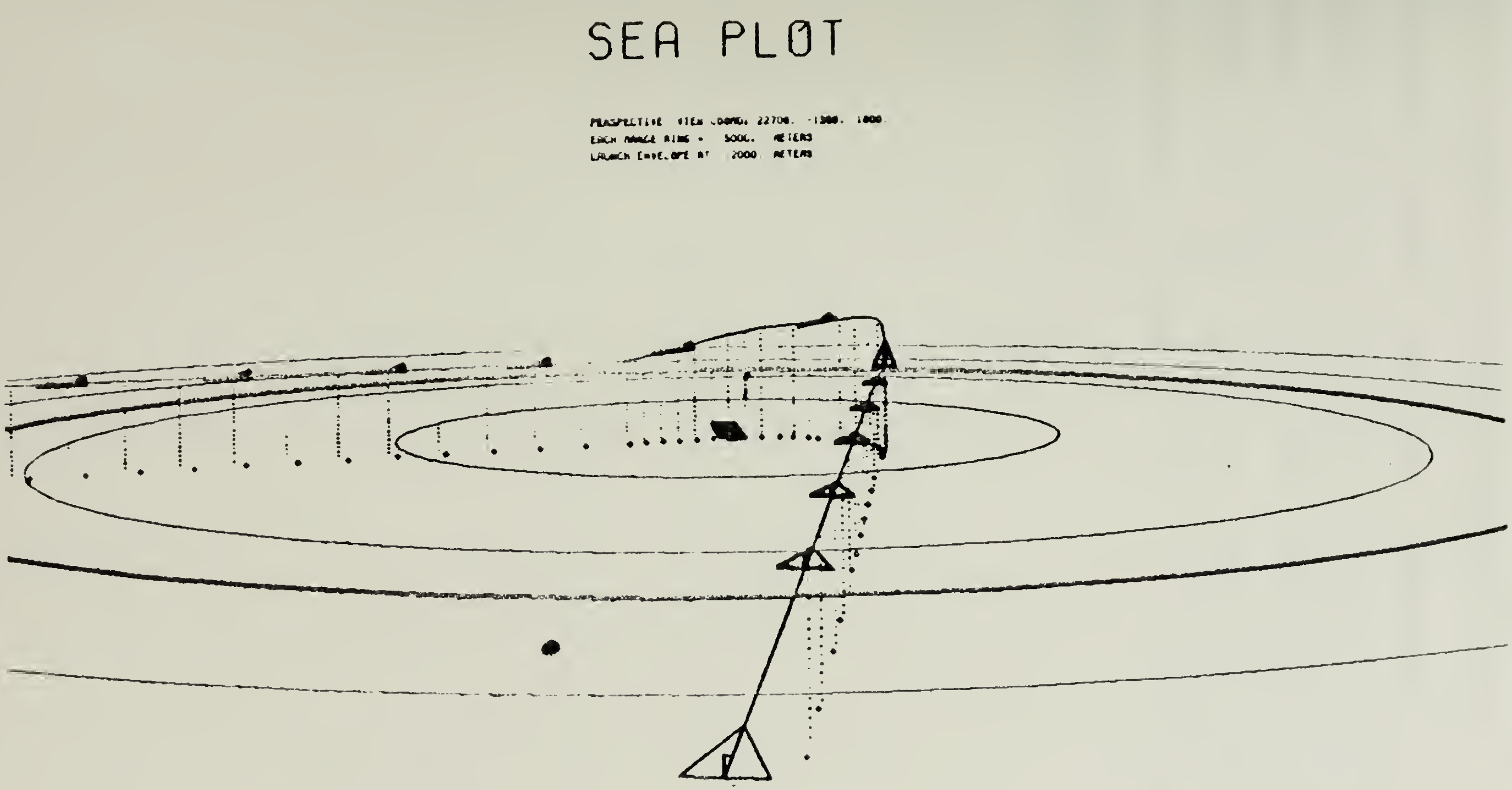

Figure 1-3: SEA PLOT Output with SCALE=3

Detailed discussion of SEA PLOT techniques and use follow in Chapters II and II which cover the conic projection and program input and organization, respectively. Appendix A provides a complete program listing, Appendix B contains a listing of the conic projection routines, and Appendix C discusses use of the Tektronix developed interactive graphics software PLOT-80 on the Tektronix 4081 graphics terminal.

II. CONIC PROJECTION

This chapter discusses the projection technique employed by SEA PLOT to provide a perspective two-dimensional display of an object in three-dimensional space. The conic projection technique was selected for SEA PLOT from among a multitude of algorithms [Ref. 1]. This technique is particularly well suited to the purpose of displaying a three-dimensional object as if it were being physically viewed by the user's eye in that not only is the solid body orientation in space accurately portrayed, but perspective depth and distance from the viewer's eye are preserved. Thus, it becomes an excellent tool for viewing an aircraft-ship engagement as observed from a distant location.

As with any effort to display a solid object using a mathematical definition of body doorcinates, the elimination of hidden lines and surfaces as they are cloaked from view by surfaces in a more prominent viewing position presents a problem to the programmer. Moreover, the computation time required to mask hidden surfaces grows with the square of the situation complexity [Ref. 2]. However, the use of simplistic solid body models as opposed to detailed body structures eliminates the need for a comprehensive hidden surface removal routine. Since SEA PLOT is intended to

provide a symbolic overview rather than a definitive display of an attack scenario, this technique has been adopted. Consequently, the aircraft body is displayed in triangular form approximating the wing planform of a delta wing aircraft. The triangle is elongated with an included tail section to enhance the perspective orientation of the aircraft in three-space. Likewise, the target ship is projected as a two-dimensional hull form oriented along a user's input target heading. However, should future use of SEA PLOT dictate more comprehensive target and attacker models, Reference 3 characterizes ten hidden surface algorithms which a programmer may employ.

The conic projection technique involves placement of the viewer's "eye" in an earth-fixed coordinate system. Associated with the positioning of the eye is a cone of vision encompassing the field of view and a projection plane onto which the scenario is to be displayed. SEA PLOT automatically determines these positions based upon the input flight profile of the attacking aircraft. The viewing plane is placed 2000 meters beyond (as viewed from the origin) the attacker's initial flight path point, oriented perpendicular to a line drawn through the earth-fixed origin (the location of the ship) and the initial attacker position. The eye coordinates are placed 2000 meters beyond the viewing plane along the same line. This automatic placement of the viewing system eliminates the

requirement for the user to determine an optimal viewing point which, if ill-placed, could result in plot distortion should the field of view exceed the cone of vision. Should another viewing position relative to the attacker's air track be desired, Chapter III discusses the procedure to alter the perspective position. Appendix B contains the actual routines employed by SEA PLOT to accomplish the conic projection and can be used as a "skeleton program" to be applied to three-dimensional graphics endeavors of some other nature.

The conical projection method described in this chapter is essentially the same as that used in some of the displays with the Differential Maneuvering simulator at the NASA Langley Research Center [Ref. 4]. It is summarized in Reference 5, from which the following derivations were taken. Reference 5 also discusses additional applications of the technique.

First consider an aircraft at some distance from the center of an earth-fixed coordinate system. The coordinate system, with axes labeled Xe, Ye, Ze is right handed, the Xe-Ye plane tangential to the earth at the origin, with the Z axis pointing up as shown in Figure 2-1.

The solid body (in this case the aircraft) center of gravity (origin of the aircraft body-fixed coordinates) is then positioned at $\overline{X}e$, $\overline{Y}e$, $\overline{Z}e$, with the body orientation in three-space given by the Euler angles ψ, θ, ϕ corresponding

to heading, pitch, and roll, respectively. The aircraft body axes (Xb, Yb, Zb) are chosen such that Xb points from the origin toward the nose, with the Yb axis pointing in the direction of the left wing which gives a conventional right-handed system with the Zb axis pointing up from the aircraft center. Coordinates of the aircraft body are given by Xp(i), Yp(i), Zp(i) which are relative to the aircraft center of gravity. These body coordinates can then be transformed to earth-fixed coordinates Xp'(i), Yp'(i), Zp'(i) using

$$\begin{bmatrix} X'p(i) \\ Y'p(i) \\ Z'p(i) \end{bmatrix} = [D] \begin{bmatrix} Xp(i) \\ Yp(i) \\ Zp(i) \end{bmatrix} + \begin{bmatrix} \overline{X}e \\ \overline{Y}e \\ \overline{Z}e \end{bmatrix}$$

where $\left[D\right]$ is the direction cosine matrix from earth-fixed axes to body axes:

$$[D] = \begin{bmatrix} \cos\theta \cos\psi & \cos\theta \cos\psi & \sin\theta \\ -\cos\psi \sin\theta \sin\phi + \sin\psi \cos\phi & \cos\psi \cos\phi - \sin\psi \sin\theta \sin\phi & \cos\theta \sin\psi \\ -\cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi & -\sin\psi \sin\theta \cos\varepsilon - \cos\psi \sin\phi & \cos\theta \cos\phi \end{bmatrix}$$

After transforming the aircraft body coordinates to earth-fixed coordinates, the viewing plane must be established onto which the conic projection will be made. In so doing, it is convenient to establish a new coordinate system Xev, Yev, Zev, with the origin at the eye point (Xeye, Yeye, Zeye). The X axis of this coordinate system extends from the eye point to some view point located on the viewing plane. This, then, represents the axis of the projection cone. The viewing coordinate system is obtained by two rotations of the earth-fixed system. First, a rotation is made about the Ze axis by the angle θev, the angle between the Xe axis and the projection of the line connecting the eyepoint and the viewpoint onto the Xe-Ye plane. The second is a pitch rotation about the new Y axis by an angle θev. This orients the cone of vision in space. From Figure 2-1, it can be seen that

$$\theta ev = \tan^{-1} \left(\frac{Zview - Zeye}{R2D}\right)$$

where

$$R2D = [(Xview - Xeye)^2 + (Yview - Yeye)^2]^{1/2} .$$

Any point in the earth-fixed coordinate system can now be expressed in the viewing coordinate system by

$$\begin{bmatrix} Xev \\ Yev \\ Zev \end{bmatrix} = [A] \begin{bmatrix} X'p(i) \\ Y'p(i) \\ Z'p(i) \end{bmatrix} - \begin{bmatrix} Xeye \\ Yeye \\ Zeye \end{bmatrix}$$

where [A] is the previously defined [D] matrix with

$$\begin{aligned} \theta &= \theta ev \\ \psi &\quad \psi ev \\ \phi &\quad \phi ev \end{aligned}$$

The final step, after transforming the body coordinates of the solid object to viewing coordinates is to make a two-dimensional projection of the three-dimensional object onto a flat picture plane. (For SEA PLOT, this picture plane is the plotting paper; but more generally, it would be the screen of the graphics display console.) To do this, the distance from the viewer (user's eye) to the picture plane is defined as the distance XPLANE. (See Figure 2-2.) This permits determination of the two dimensional plotting coordinates [YSCOPE, ZSCOPE) for the projection of a three-dimensional object onto a two-dimensional plotting plane. From Figure 2-2, it can be seen that the two triangles $OS\bar{P}$ and OP"P' are similar. Therefore,

$$\frac{XPLANE}{Xev} = \frac{YSCOPE}{Yev}$$

or

$$\boxed{YSCOPE = \frac{Yev}{Xev} \times XPLANE}.$$

Also similar triangles are $O\bar{P}\bar{P}'$ and OPP'. Therefore,

$$\frac{ZSCOPE}{Zev} = \frac{O\bar{P}}{OP'}$$

From the previous set of similar triangles, it follows that

$$\frac{O\bar{P}'}{OP'} = \frac{XPLANE}{Xev} \quad .$$

Hence,

$$\boxed{ZSCOPE = \frac{Zev}{Xev} \times XPLANE}$$

Appendix B provides the FORTRAN coding to implement these derivations.

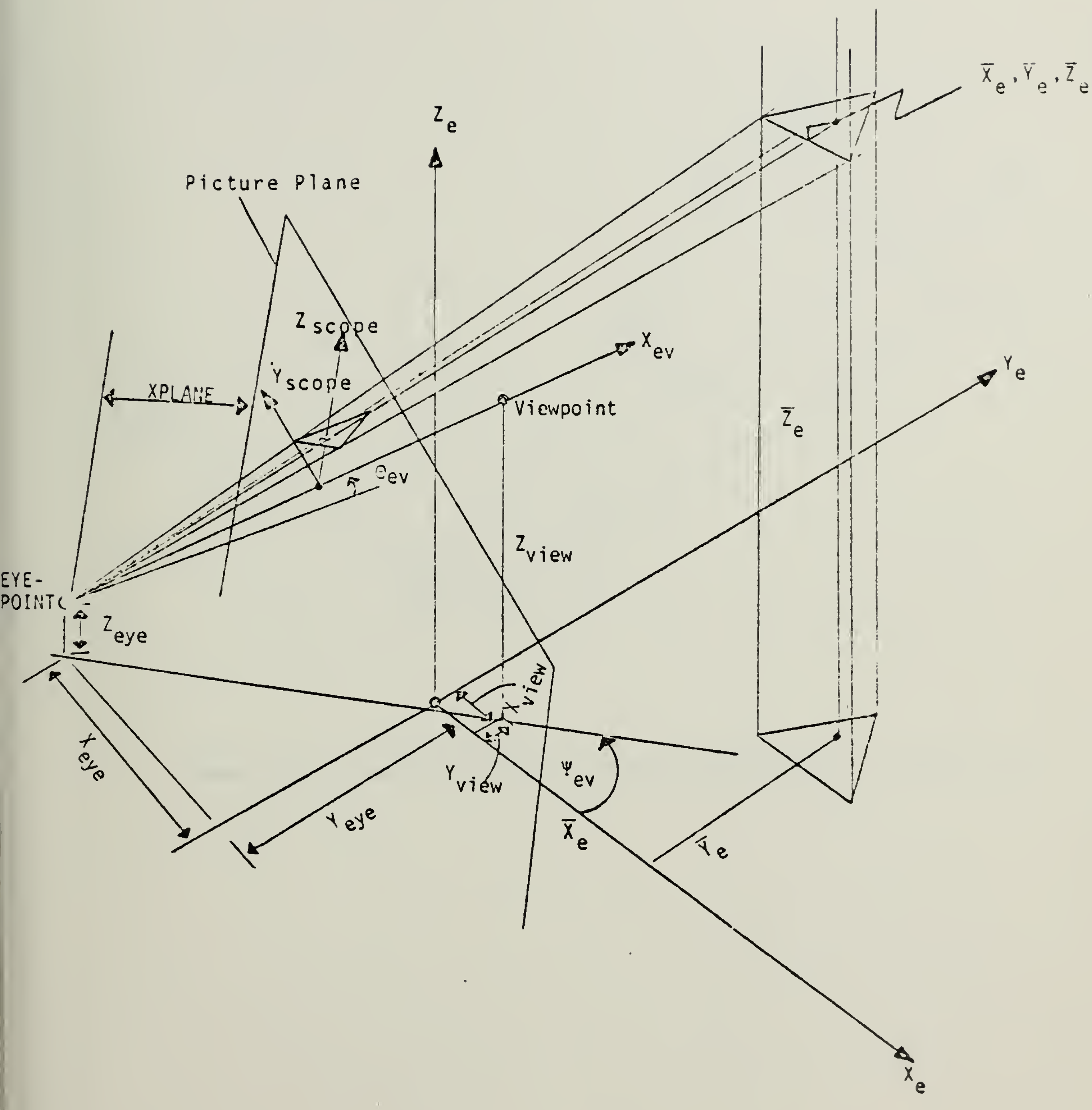

Figure 2-1: Conical projection of aircraft image on to picture plane.

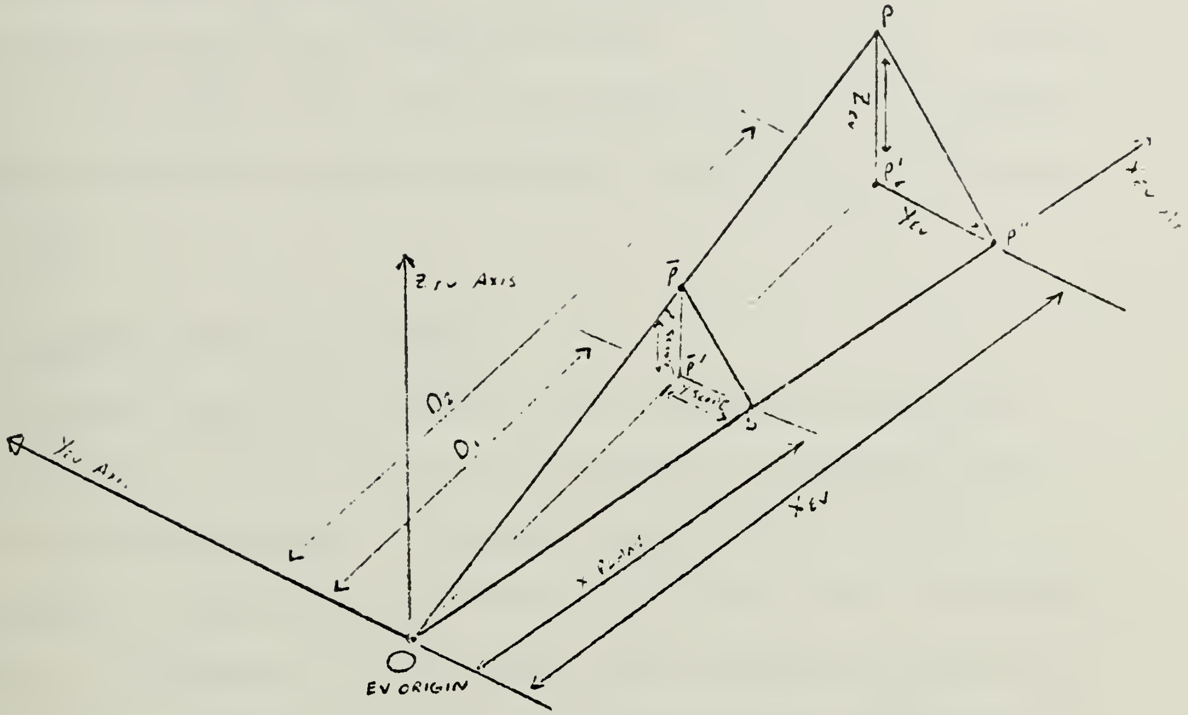

Figure 2-2: Transformation from the EV system into the SCOPE system.

III. PROGRAM INPUTS AND ORGANIZATION

This chapter discusses SEA PLOT required input data and the organization of the program itself. It provides information on the input data and input format as well as detailed discussion of the thirteen different routines that comprise SEA PLOT.

A. PROGRAM INPUTS AND FORMAT

Program inputs for SEA PLOT are intended to provide the user with a wide variety of options in designing the desired display output. SEA PLOT inputs fall into two categories: plotting parameters and flight path coordinate points. A maximum of 200 flight path coordinate points may be plotted. Following is a description of input elements and format.

1. Card 1 - Plotting Parameter Input Card

a. Col 1-3: Enter integer number of flight path milestones, right justified.

b. Col 4: Enter integer value of the type of ship defensive missile envelope. The two types of envelope display available are a solid cylinder (Figure 3-1) or a bold black range ring drawn on the sea surface (Figure 3-2). Integer inputs are:

1 - range ring.

0 - solid cylinder.

c. Col 5: Enter integer value indicating if a plot of the coordinate axis is desired as follows:

1 - axis plot is desired (Figure 3-3).

0 - no axis plot desired.

d. Col 6-15: Enter decimal value corresponding to the desired plot scale. Figures 1-1, 1-2, 1-3 show representative scaled plots corresponding to SCALE=1.,2.,3., respectively. This input parameter essentially magnifies the plot with the center of magnification at the origin of the earth-fixed coordinate system. No distortion of the perspective view results from increased scale factors.

e. Col 16-25: Enter floating point ship defensive missile range in meters.

f. Col 26-34: Enter floating point value of target heading in degrees, counter clockwise from the x axis.

2. <u>Cards 2 - 201 (One Card per Milestone)</u>

a. Col 1-10: Floating point value of aircraft x coordinate in meters.

b. Col 11-20: Floating point value of aircraft y coordinate in meters.

c. Col 21-30: Floating point value of aircraft z coordinate in meters.

d. Col 31-40: Floating point value of aircraft roll in degrees.

e. Col 41-50: Floating point value of aircraft pitch in degrees.

f. Col 51-60: Floating point value of aircraft heading in degrees.

B. PROGRAM ORGANIZATION

SEA PLOT consists of thirteen different routines which can be categorized as plot management routines, graphics routines, conic projection routines, and plotting routines. This organization is intended to provide a smooth flow of data from one program element to another as well as to present a reasonably coherent program structure to the user. Sufficient commentary data has been included in the program to facilitate future changes or alterations in as uncomplicated a manner as possible. Program comments, coupled with the detailed discussion in this chapter, should provide sufficient program background such that program modifications to meet a variety of needs could be relatively expeditiously made. However, a word of caution is necessary when altering data or variable elements. Much data is passed from routine to routine in the form of Common Blocks. While this technique provides an efficient data flow, it does detract from the comprehensibility of the coding. Consequently, altering data in one subroutine could have unpredictable consequences if it impacts on calculations taking place in other subroutines. The majority of the information passed in common involves input data arrays and scaling parameters, some from external sources and some

internally generated. Before altering these variables, one should thoroughly understand their uses in the various algorithms in which they are employed.

1. Plot Management Routines

These routines consist of those algorithms required to input data, determine plotting scales, label plots, and set up the eye-view coordinate system. Three routines comprise this section; they are MAIN, SEAPLT, and AXIS. Each of these is described below.

a. MAIN Program

MAIN is straightforward in that its primary purpose is to read the input data in the format previously described. It also establishes the common blocks "AIR," "PARAM," and "INPUT." Variables and their descriptions are as follows:

(1) XAIR, YAIR, ZAIR - Floating point arrays of 200 elements each which contain the attacking aircraft x, y, z coordinates, respectively.

(2) ACROLL, ACPTCH, ACHEAD - Floating point arrays of 200 elements each which contain the aircraft roll, pitch, and heading corresponding to each coordinate position.

(3) MDISP - Integer input value determining the type of missile envelope display to be plotted. MDISP=1 causes a single bold range ring to be plotted on the surface (z=0). MDISP=0 causes a three-dimensional

cylinder to be plotted. The height of this cylinder corresponds to twice the maximum aircraft input altitude. The radius of the envelope is the missile slant range which is an input variable. (See Figures 3-1 and 3-2.)

(4) IAXIS - An integer input value used to determine if a plot of the coordinate axes is desired. IAXIS=0 indicates no axes are to be drawn. IAXIS=1 causes the coordinate axes x, y, z to be displayed.

(5) SCALE - An input floating point value used to determine the magnification factor to be applied to the plotted output. This permits the user to "zoom in" on the center of the plot, magnifying it by the value of SCALE. Figures 1-1, 1-2 and 1-3 demonstrate the use of this parameter.

(6) RNGMIS - An input floating point variable used to define the maximum range of the ship's defensive missiles.

(7) THEAD - An input floating point variable corresponding to the heading of the ship target measured in degrees from the axis.

Printed output from MAIN is a list of input variable values and aircraft flight path data.

b. SEAPLT

SEAPLT is the driving routine for the entire program. It searches for maximum flight path values, positions the eye and viewplane coordinates, and sequences

through the plotting routines to cause the graphic output. The viewing plane is positioned 2000 meters behind the initial flight path position along a three-dimensional line from the origin through the initial position. The center of the eye-view coordinate system is placed on this same line 2000 meters beyond the viewing plane. Because of the low aircraft altitudes employed in current war-at-sea tactics, this routine performs an altitude enhancement which adjusts aircraft input altitudes such that the maximum plotted altitude corresponds to the viewing height. This preserves a perspective in which the viewer is always looking along or down upon the flight path. In that all of the flight path altitudes are increased by this same value, no plot distortion occurs, overall perspective is maintained, and altitude maneuvers are enhanced.

Variables used in SEAPLOT are as follows:

(1) XVIEW,YVIEW,ZVIEW - x, y, z coordinates of the center of the viewing plane.

(2) XEYE,YEYE,ZEYE - x, y, z coordinates (earth-fixed) of the eye position. This becomes the origin of the eye-view coordinate system.

(3) XPLANE - Physical distance from the viewer to the plotting surface. This variable is particularly germaine to plotting on a graphics terminal. Here it is fixed at 30 cm.

(4) DX - Separation of range rings from the plot origin. DX is currently fixed at 5000 meters.

(5) BESTZ - Altitude enhancement factor applied to the input aircraft flight profile altitude.

(6) XMAX - Maximum aircraft x coordinate.

(7) YMAX - Maximum aircraft y coordinate.

(8) ZMAX - Maximum aircraft z coordinate.

(9) THETA - Angle in radians of the line drawn through the origin and the first aircraft coordinate position.

c. AXIS

In addition to drawing coordinate axes, subroutine AXIS also performs plot labeling and display scaling. Consequently, this subroutine must be called whether or not a plot of the coordinate axes is desired. The only parameter required by AXIS is NFLAG. This is the IAXIS argument previously described.

Scaling performed by AXIS is accomplished in such a manner as to center the plot on a Versatec 20-inch by 20-inch plotting sheet. To do this, AXIS first projects a circle of radius corresponding to the length of a vector drawn from the earth-fixed origin out to the point given by the coordinates (XMAX,YMAX). This circle consists of 180 data points. After calling the conic projection routines to make the projection circle, AXIS searches for the longest vector from the origin out to the periphery of the circle. Since the projected circle will appear as an ellipse whose elongation

will depend on the viewing angle, this longest vector will vary from view to view. Once this longest projection vector is known, AXIS reorigins the plot to the center of the plotting sheet at plotter coordinates (10,10), then scales the length of the longest projection vector, named SFACT, such that its physical size is 10 inches on the plotter. This permits the development of an over-all plot scaling factor XSCALE where:

$$XSCALE = 10.0/SFACT$$

Plot magnification can be accomplished to meet the user's specifications by multiplying XSCALE by the input variable SCALE. This, of course, will result in the ultimate computation of many vectors whose length will exceed the dimensions of the plotting paper. However, these vectors will be clipped at the edge of the plotting sheet and will not degrade the plotted output; though a diagnostic, which may be ignored, will be generated on the ouptut listing containing the flight path data.

The final function of AXIS is to label the plot. The labeling instructions in the program are software dependent. Those in AXIS are written specifically for VERSATEC software. Hence, careful attention must be paid to this routine when adapting it to another graphics system.

Axis options and labeling are shown in Figure 3-3 which shows the axis output.

2. Graphics Routines

The graphics routines described in this section are those elements which accomplish the actual pen moves required to draw lines or dots on the plotting sheet. Thus, these routines are highly software dependent and represent the major changes required to adapt SEA PLOT to another graphics system. SEA PLOT requires only three types of graphics operations: drawing solid lines, drawing vertical dashed lines, and plotting dots. The two subroutine descriptions which follow discuss these operations.

a. Subroutine DRAW

This subroutine causes an array of projected coordinate data points to be connected either by a solid line or to be plotted as dots on the plotting sheet. Up to 200 data points may be plotted. Projected coordinates are first scaled to a 20-inch by 20-inch plot size using the XSCALE factor developed in AXIS. They are then shifted from the earth-fixed coordinate center to the plotting sheet coordinate center by adding a value XORIG, or YORIG which corresponds to the displacement of the projected earth origin from the plotter origin. DRAW next causes a pen up move to be made to the initial array position and then either draws a line connecting the points or plots a dot corresponding to each coordinate position.

Variables used in DRAW are:

(1) X,Y - 200 element input arrays of projected data to be plotted.

(2) XPLOT, YPLOT - 200 element arrays containing the scaled projection data to be plotted.

(3) N - Integer value indicating the number of coordinate pairs to be plotted.

(4) ITYPE - Integer value which determines the type of plot to be drawn. ITYPE=0 causes dots to be plotted. ITYPE=1 causes solid lines to be drawn.

b. Subroutine VDASH

Subroutine VDASH causes a vertical dashed line, parallel to the z axis, to be drawn between two points whose coordinate values are passed as arguments. It should be emphasized that only a vertical line is drawn. Should the two points to be connected by the line not be vertically positioned, the routine will draw a dashed line from the first point up to the Y value of the second point. As with subroutine DRAW, all data values are scaled to the XSCALE factor generated in AXIS.

Variables used in VDASH are:

(1) X1,Y1 - Projection coordinates of the lower data point.

(2) X2,Y2 - Projection coordinates of the upper data point.

(3) DY - vertical separation between the data points.

3. <u>Conic Projection Routines</u>

The three subroutines that follow are those routines used to implement the conic projection technique derived in Chapter II. They are also reproduced in Appendix B to serve as a ready reference "skeleton program" to be used in other graphics efforts.

a. Subroutine MULT

MULT performs a matrix multiplication of a square 3 x 3 matrix with a column vector, returning the resultant column vector to the calling program. The operation performed is

$$\begin{bmatrix} Xnew \\ Ynew \\ Znew \end{bmatrix} = [D] \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

where the vector[X,Y,Z] and the matrix [D] are passed to the subroutine as arguments.

b. Subroutine DIRCOS

DIRCOS is a rather simplistic routine which merely sets up the direction cosine matrices discussed in Chapter II. Inputs to the routine are the Euler angles psi, theta, and phi corresponding to solid body roll, pitch, and yaw in three-space. DIRCOS output is the three-dimensional matrix [D] or [A] employed by the projection. Also available as an output is the transpose of [D]. In that the coding used

in DIRCOS is a straightforward assignment of variables to matrix positions, no further discussion will be attempted here.

c. Subroutine CONPRO

CONPRO is the primary routine for the conic projection scheme implementing the derivations in Chapter II. Inputs to CONPRO are the center of gravity coordinates of the solid body to be projected, the number of defined body coordinates, the array of body coordinate points, and the body Euler angles in degrees measured counterclockwise from the earth-fixed X axis. After performing all translations and rotations required by Chapter II, CONPRO outputs the plotting surface coordinates of the projected solid body. These coordinates are passed by the calling program to the DRAW routine where they are scaled and plotted.

Variables used in CONPRO are:

(1) CG - Three element array containing the X,Y,Z earth-fixed coordinates of the solid body center of gravity.

(2) NPOINT - Integer value indicating the number of data points used to define the solid body.

(3) XP - Array of body X coordinate values in the body-fixed reference system. This array is variably dimensioned to NPOINT elements.

(4) YP - Array of body Y coordinate values in the body-fixed reference system. This array contains NPOINT elements.

(5) ZP - Array of NPOINT body Z coordinate values in the body-fixed reference system.

(6) EULER - Three element array containing the Euler angles for roll, pitch, and heading in degrees.

(7) YSCOPE - Output array of NPOINT projected plotter X coordinates.

(8) ZSCOPE - Output array of NPOINT projected plotter Y coordinates.

4. Plotting Routines

The five subroutines that follow are those algorithms which develop the different picture segments that comprise the SEA PLOT output. The five elements displayed are: the circular surface grid consisting of 5000-meter range rings, the aircraft ground and air tracks, the aircraft body plot, the target body plot, and the defensive missile launch envelope. Since more sophisticated graphics languages such as PLOT 80 employ the concept of graphics control blocks (discussed in Appendix C) or picture segments, SEA PLOT was organized along these same lines to facilitate easier transition to one of these more powerful software systems. Using this scheme, each subroutine will cause a specific picture segment to be displayed as directed by the sequencing routine SEAPLT. As a result, the user has the option of deleting those picture elements which are not desired. For example, by circumventing the call to the target subroutine,

one could get a plot of only the surface grid and aircraft flight profile without a target being shown.

a. Subroutine CGRID

Subroutine CGRID projects and draws a series of concentric range circle centered at the target position at the origin. As presently written, these range circles are drawn at 5000-meter increments as indicated by the variable DX discussed in SEAPLT. Each range circle is defined by plane polar coordinates consisting of 181 data points per circle. The number of circles drawn is a function of the maximum X coordinate for the aircraft flight profile such that the number of circles is the quotient of the maximum X range and the radius increment DX.

Variables used in CGRID are:

(1) X,Y,Z - 181 element arrays containing the unprojected coordinates of the data points defining each range ring. The Z array is set to zero corresponding to zero altitude.

(2) YSCOPE,ZSCOPE - 181 element arrays containing the projected plotter coordinates of the range circle data points. These arrays are passed to the DRAW subroutine where they are scaled and plotted.

(3) DX - Interval between range circles in the earth-fixed reference system.

(4) RADIUS - Radius of each range circle in the earth-fixed system.

Figure 3-4 portrays the output from CGRID.

b. Subroutine ACPLOT

Subroutine ACPLOT plots up to 200 aircraft flight path coordinate points in a three-dimensional perspective view. To enhance the view, both ground and air tracks are displayed. The ground track is plotted by setting the aircraft Z coordinate to zero, then projecting the X,Y,Z coordinates and plotting the data as series of dots corresponding to each input point. In a like manner, the air track is projected without zeroing the Z coordinate. A solid, bold line connects all air track data points. To reinforce the altitude perspective, an optional dashed vertical line is drawn between corresponding air and ground track data points. This vertical line option parameter is a program change option which is specified by the calling program not by a user input.

Aircraft maneuvers in roll, pitch, and yaw are displayed by plotting the aircraft body at positions corresponding to every 2000 meters of ground travel in the earth-fixed system. This is accomplished by a check of ground distance traveled between input points followed by a call to subroutine ACRAFT after 2000 meters of displacement has been detected. Because the surface area of the aircraft body is insignificant with respect to the total surface area displayed by the plot, a magnification factor is applied to the aircraft body. It has been experimentally determined

that a magnification factor of 15 permits an optimal body display.

Inputs to ACPLOT include the number of milestones to be plotted, the vertical line option parameter, and magnification factor. These and the other internal variables used in this routine are discussed as follows:

(1) N - Integer number of aircraft milestones to be plotted.

(2) VLINE - Decimal flag used to exercise the dashed vertical line option. If VLINE=1, the vertical altitude lines are drawn. Otherwise, no vertical lines are drawn between tracks.

(3) AMAG - Decimal magnification factor to be applied to the aircraft body plot.

(4) X,Y,Z - 200 element arrays containing the earth-fixed center of gravity coordinates (input milestones) for the body plot and the flight profile tracks.

(5) XPLOTA,YPLOTA - 200 element arrays containing the projected X,Y coordinates for the air track.

(6) XPLOTG,YPLOTG - 200 element arrays containing the projected X,Y plotter coordinates for the ground track.

Figures 3-5 and 3-6 show the plot output options for ACPLOT.

c. Subroutine ACRAFT

ACRAFT contains the predefined aircraft body coordinates in the body-fixed coordinate frame. This routine applies the magnification factor to the body coordinates, causes a projection of the body to be made and a plot output to be generated. The actual body coordinates are defined in data statements in the program and are not user verifiable.

Inputs to ACRAFT are the Euler of angles roll, pitch and yaw, the center of gravity coordinates in the earth-fixed reference, and the magnification factor. These and other variables used by ACRAFT are defined as follows:

(1) EULER - A three element array containing the body roll, pitch, and yaw angles in degrees.

(2) CG - A three element array containing the X,Y,Z earth-fixed coordinates of the center of gravity.

(3) XAC, YAC, ZAC - Eight element arrays containing the predefined aircraft body coordinates in the body reference frame.

(4) AMAG - Magnification factor to be applied to the body coordinates.

The output from ACRAFT can be seen in Figures 3-5 and 3-6.

d. Subroutine TARGET

TARGET performs essentially the same function as ACRAFT except that it contains the predefined body coordinates of the ship target. Because an excessive number of plotted

vectors tends to make the overall plot output too busy, the target is displayed as a hull planform only. The planform is shaded to accentuate its position on the plot. At present, the target is plotted at the earth-fixed origin. However, this is not a rigid requirement since this particular routine has been written in a general manner to permit positioning the target anywhere in the viewing area. Thus, the location of the target center of gravity is an input from SEAPLT, the calling program, and can be altered to fit the user's needs. The other input to TARGET is the ship heading measured in degrees from the earth-fixed X axis. This also is a user specified input parameter.

Variables used in TARGET are:

(1) CG - A three element array containing the target earth-fixed center of gravity coordinates.

(2) THEAD - Target heading measured in degrees counterclockwise from the earth-fixed X axis.

(3) TMAG - Magnification factor applied to the body fixed target coordinates. This variable is set to the same value as AMAG in ACRAFT.

(4) XTGT,YTGT,ZTGT - 26 element arrays containing the predefined body coordinates of the target in the body-fixed reference system.

(5) XSCOPE,YSCOPE - 26 element arrays containing the projected coordinates of the target in the plotter reference system. These are scaled and plotted by a call to DRAW.

Figures 3-7 and 3-8 demonstrate the plot options available to TARGET.

e. Subroutine LAUNCH

LAUNCH causes a plot display of the target ship's defensive missile envelope to be drawn. Two options are available. They are a three-dimensional cylinder or a bold black circle plotted on the surface as shown in Figures 3-1 and 3-2, respectively. Selection of the desired envelope plot is a user input parameter. Inputs to LAUNCH are the missile range and the display type.

The cylindrical display is defined using cylindrical coordinates such that the height of the cylinder cooresponds to twice the maximum aircraft input altitude. Cylinder construction consists of ten circles vertically separated by a distance of one fifth the maximum aircraft altitude. Each circle consists of 180 dots.

The range ring display merely consists of 180 coordinate points expressed in plane polar coordinates. The radius of the circle corresponds to the maximum missile launch range.

Variables used in LAUNCH are:

(1) RANGE - Maximum missile launch range. This is an input parameter from the calling program (SEAPLT).

(2) IDISPL - Integer display option flag such that IDISPL=1 causes the single range ring display. Otherwise the cylinder display is drawn.

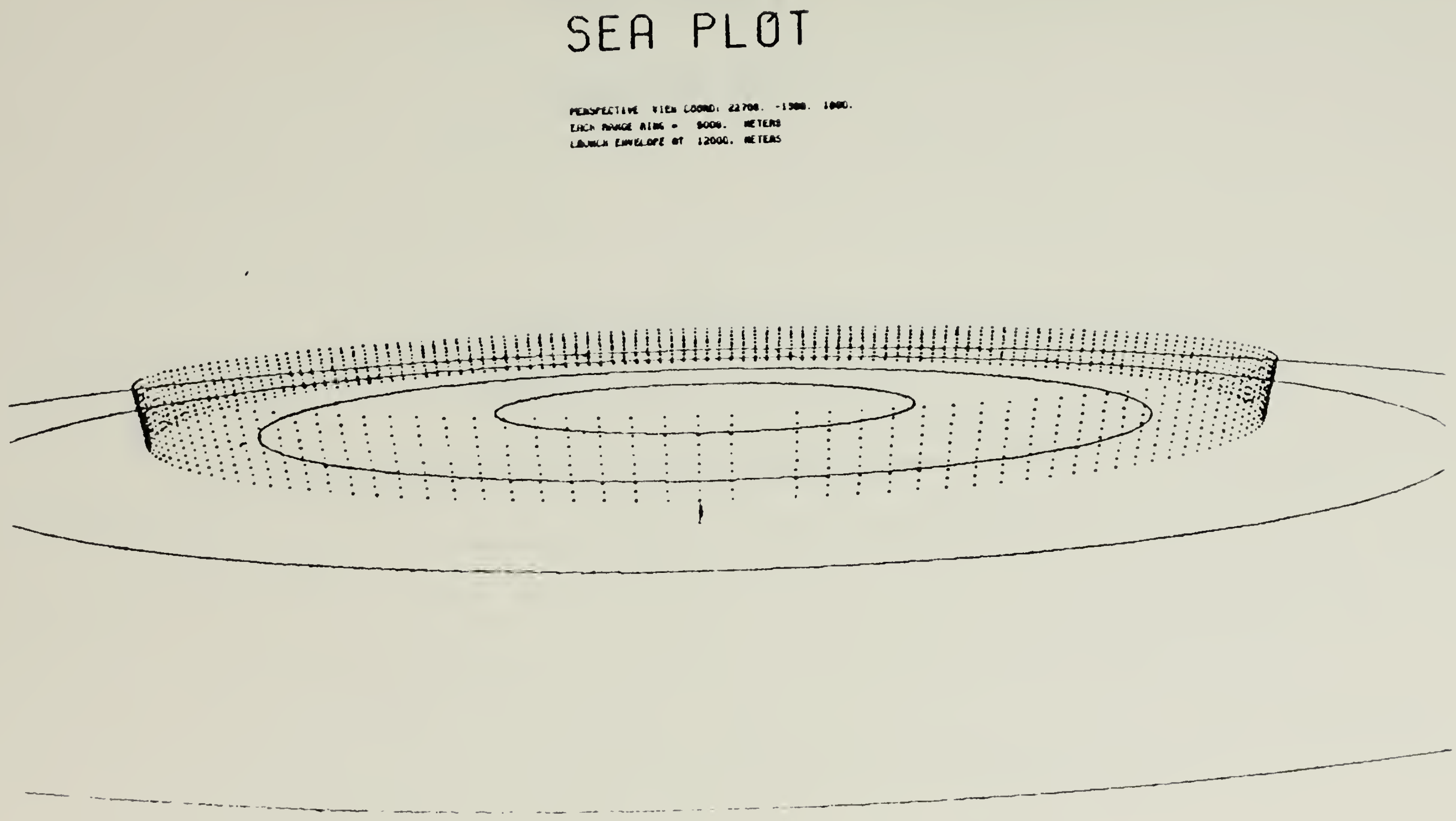

Figure 3-1: Cylinder Launch Envelope Display

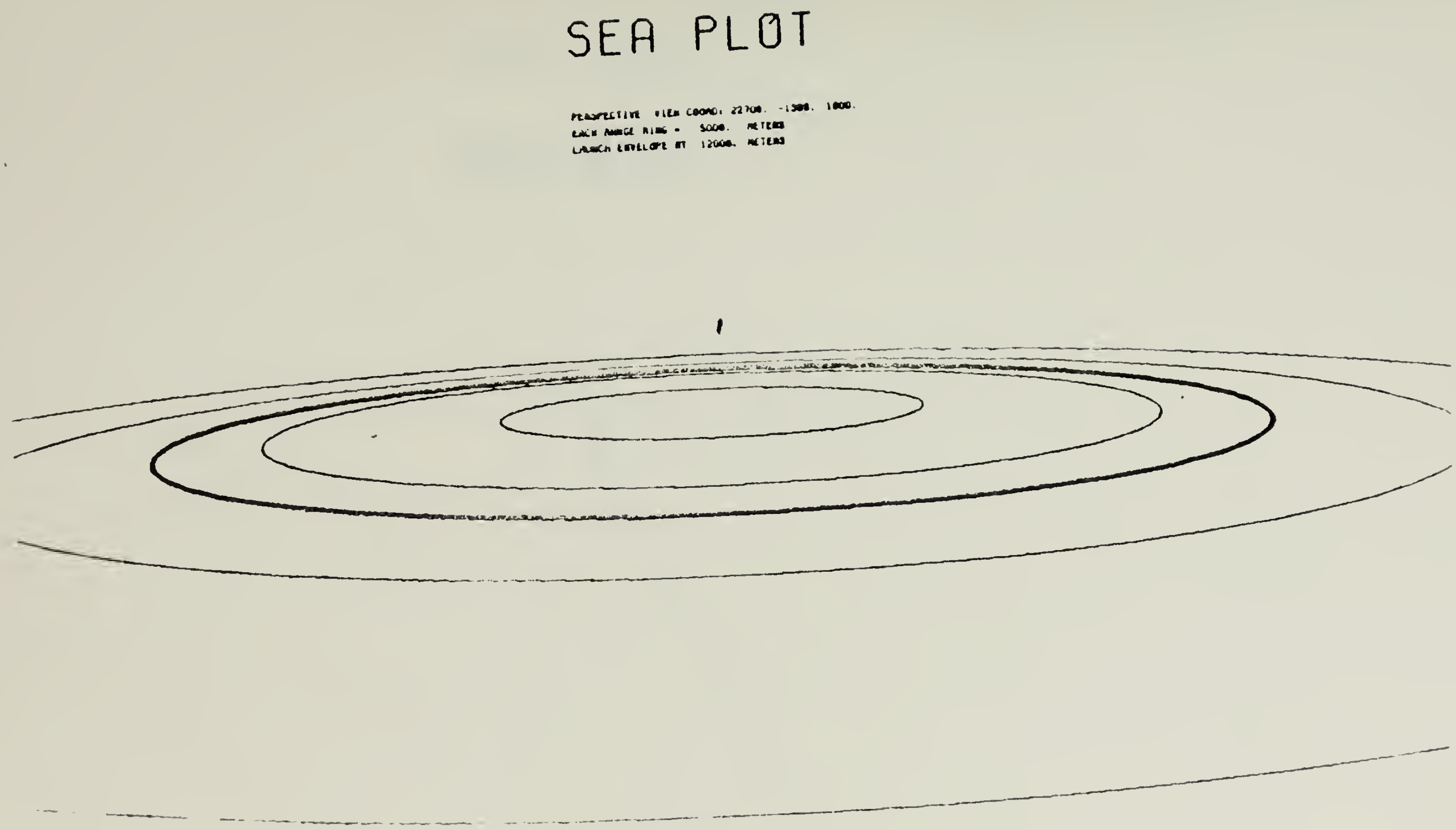

Figure 3-2: Range Ring Launch Envelope Display

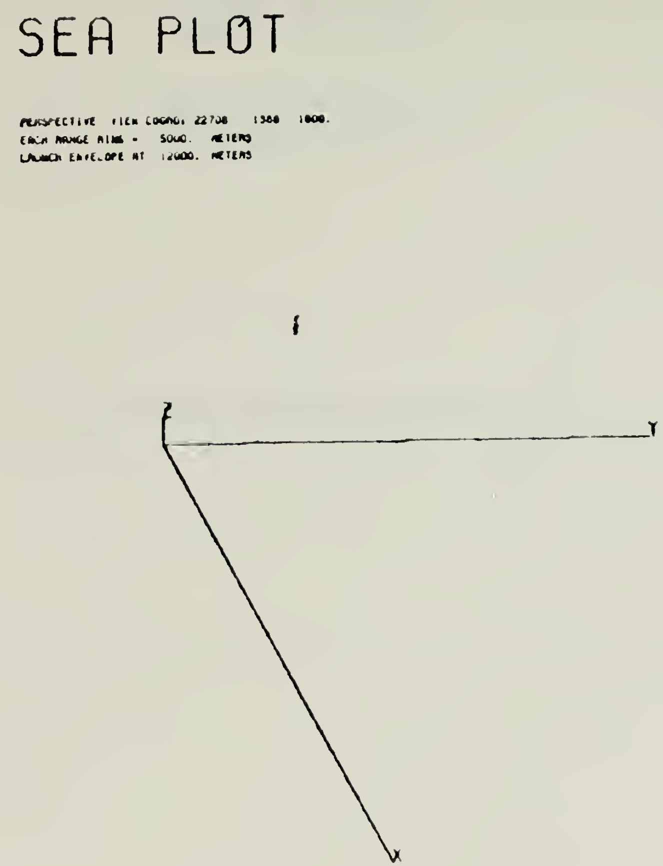

Figure 3-3: Coordinate Axis Display Option Output from Subroutine AXIS

Figure 3-4: CGRID Range Ring Output

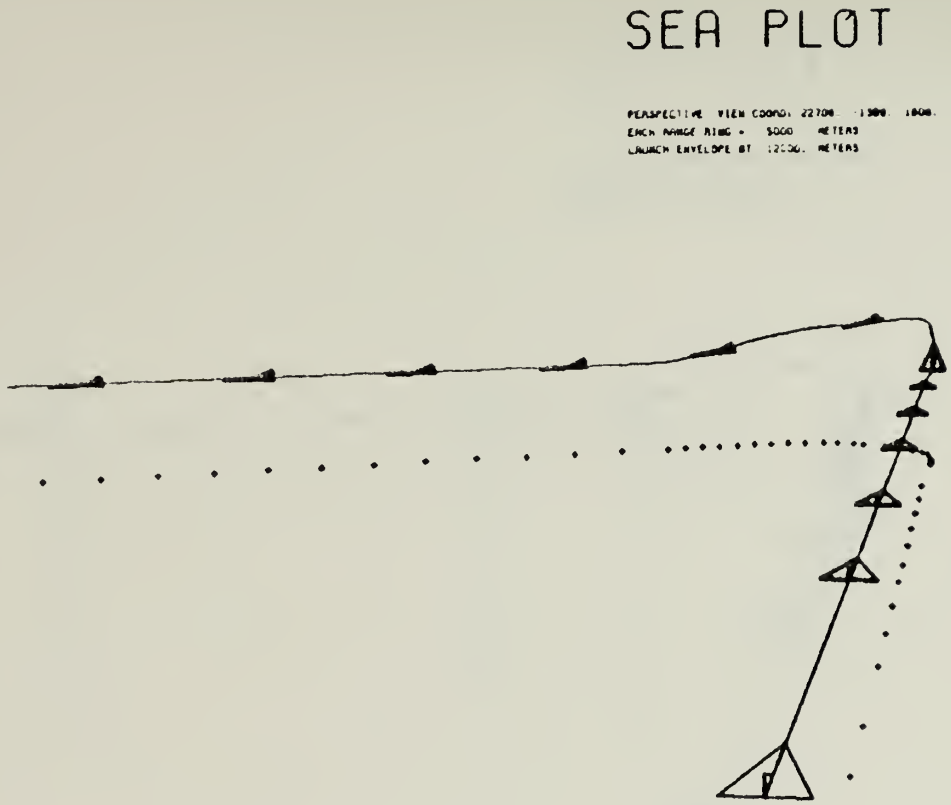

Figure 3-5: ACPLOT Output without Altitude Enhancement Lines

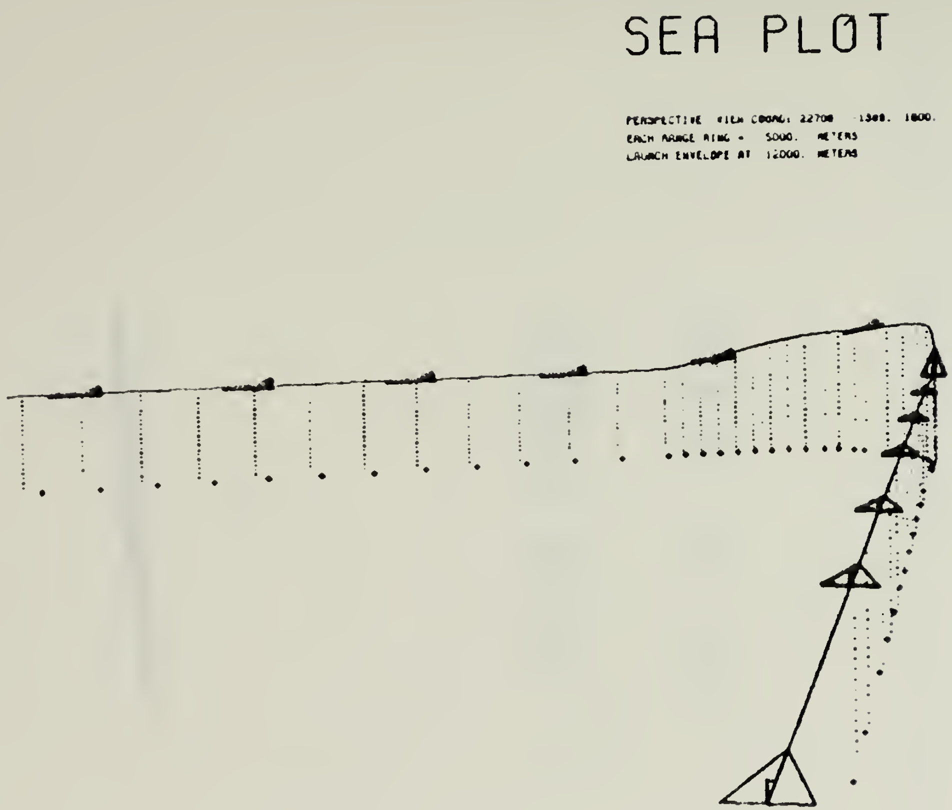

Figure 3-6: ACPLOT Output with Altitude Enhancement Lines

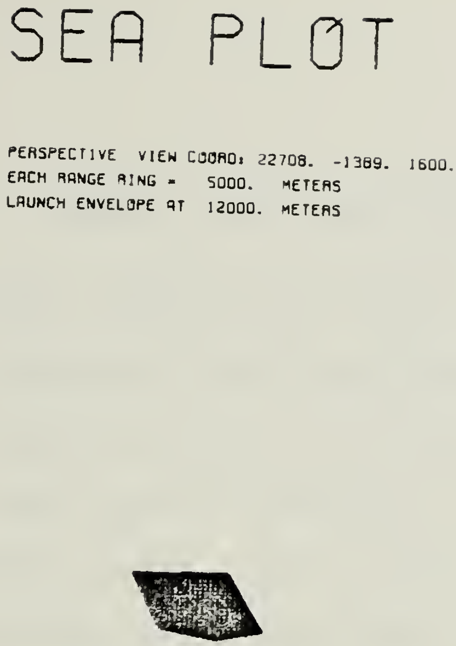

Figure 3-7: TARGET output. Ship heading=000

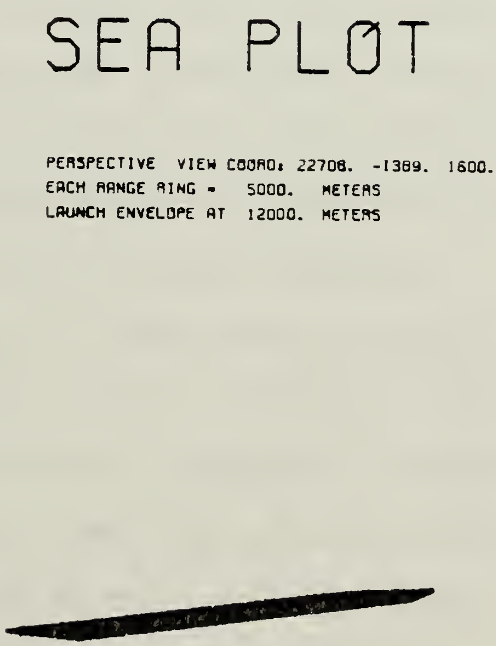

Figure 3-8: TARGET output. Ship heading=135

IV. SUMMARY AND CONCLUSIONS

SEA PLOT is a three-dimensional computer graphics program written to display a war-at-sea encounter between a single ship and a single aircraft, in which the ship is assumed to be the target and is located at the origin of an earth-fixed coordinate system. Roll, pitch, and yaw maneuvers of the attacking aircraft are shown along a three-dimensional flight path which depicts the attacker's coordinate positions in the earth-fixed reference system as well as the attitude of the aircraft body in three space. Up to 200 flight path coordinates can be plotted. To accomplish the projection of a three-dimensional scenario onto a two-dimensional plotting surface, a conic projection technique is used in which a viewing plane and an eye position are automatically computed based upon the attacking aircraft's initial position.

Displays available to the user include: (1) the earth-fixed coordinate axes, (2) 5000-meter circular range rings centered on the target, (3) ship defensive missile launch envelope displayed in solid cylindrical or plane circular form, (4) aircraft air track, (5) aircraft ground track, (6) aircraft maneuvers of roll, pitch, and yaw, (7) ship target body display located at the earth-fixed origin. To generate these displays, SEA PLOT was written using thirteen subroutines which fall into four functional categories: plot

management routines, graphics (vector drawing) routines, conic projection routines, and plotting (picture segment) routines. This organization was selected to allow for easy conversion from the present VERSATEC graphics software to other graphics languages such as the Tektronix developed PLOT 80.

APPENDIX A

SEA PLOT PROGRAM LISTING

```
      DIMENSION XAIR(200),YAIR(200,ZAIR(200)
      DIMENSION ACROSS(200),ACPTCH(200),ACHEAD(200)
      COMMON/AIR/XAIR,YAIR,ZAIR,ACROSS,ACPTCH,ACHEAD
      COMMON/GRID/XMIN,XMAX,YMIN,YMAX,CX,ZMAX,XORIG,YORIG
      COMMON/PARAM/MDISP,IAXIS,NPOINT
      COMMON/INPUT/SCALE,RNGMIS,THEAD
C
C.    MAIN PROGRAM  INPUTS AIRCRAFT FLIGHT PROFILE, AND DISPLAY
C.    PARAMETERS
C
C.    READ NUMBER OF FLIGHT PROFILE DATA POINTS (NPOINT), PLOT
C.    SCALE (SCALE), MISSILE ENVELOPE RANGE (RNGMIS), MISSILE
C.    ENVELOPE DISPLAY TYPE (MDISP), AXIS DISPLAY TYPE (IAXIS).
C.    PLOT OPTION DEFINITIONS AS FOLLOWS:
C.          SCALE - PERMITS MAGNIFICATION OF PLOT, EXPANDING ABOUT
C                   THE ORIGIN
C.          RNGMIS - RANGE OF SHIP DEFENSIVE MISSILE ENVELOPE
C.          MDISP - MISSILE DISPLAY DETERMINES TYPE OF LAUNCH
C.                  ENVELOPE TO DISPLAY:  1=SINGLE RANGE RING
C.                  2=CYLINDER IN 3 SPACE
C.          IAXIS - DETERMINE PLOT OF COORDINATE AXES:  0=NO AXIS PLOT,
C.                  1=PLOT AXES
C.          THEAD - TARGET SHIP HEADING
C.          NPOINT - NUMBER OF FLIGHT PATH MILESTONES
C.
      READ (5,900) NPOINT,MDISP,IAXIS,SCALE,RNGMIS,THEAD
C.    PRINT FLIGHT PROFILE PARAMETERS
      WRITE (6,903)
      WRITE (6,904) SCALE, RNGMIS,THEAD,NPOINT
```

```
            WRITE (6,905)
C.      READ AND PRINT FLIGHT PROFILE
            DO 10 I=1,NPOINT
            READ (5,902) XAIR(I),YAIR)I),ZAIR(I),ACROSS(I),ACPTCH(I),
            +ACHEAD(I)
            WRITE (6,906) XAIR(I),YAIR(I),ZAIR(I),ACROSS(I),ACPTCH(I),
            +ACHEAD(I)
        10 CONTINUE
C
C
C.      GENERATE PLOT
            CALL SEAPLT
C
C
C============FORMATS============
900         FORMAT(I3,2I1,3F10.2)
902         FORMAT(6F10.2)
903         FORMAT('1',30X,'FLIGHT PATH INPUT DATA:')
904         FORMAT('0',2X,'SCALE=',F4.2,2X,'MISSILE RANGE=',F10.2,
            +2X,'TARGET HEADING=',F3.2,2X,'NUMBER OF FLIGHT POINTS=',I3)
905         FORMAT('C',2X,'X',10X,'Y',10Z,'Z',10X,'ROLL',10X,'PITCH',
            +10X,'HEADING',/)
906         FORMAT(' ',2X,6F10.2)
C===============================
            STOP
            END
            SUBROUTINE SEAPLT
            COMMON/AIR/XAIR,YAIR,ZAIR,ACROSS,ACPTCH,ACHEAD
            DIMENSION XAIR(200),YAIR(200,ZAIR(200),CG(3)
            DIMENSION ACROSS(200),ACPTCH(200),ACHEAD(200)
            COMMON/EVIEW/XVIEW,YVIEW,ZVIEW,XEYE,YEYE,ZEYE,XPLANE
            COMMON/GRID/XMIN,XMAX,YMIN,YMAX,DX,ZMAX,XORIG,YORIG
            COMMON/PSCALE/XVAL,YVAL,XV,YV,XSCALE,YSCALE
            COMMON/PARAM/MDISP,IAXIS,NPOINT
            COMMON/INPUT/SCALE,RNGMIS,THEAD
C
```

```
C.        SUBROUTINE SEAPLT SETS UP TO PROJECT AND PLOT WAR AT SEA SCENARIO
C
C.        SEARCH FOR MAXIMUM FLIGHT PATH VALUES
              XMAX=ABS(XAIR(1))
              YMAX=ABS(YAIR(1))
              ZMAX=ZAIR(1)
              DO 10 I=2,NPOINT
              XABS=ABS(XAIR(I))
              YABS=ABS(YAIR(I))
              IF(XABS.GT.XMAX) XMAX=XABS
              IF(YABS.GT.YMAX) YMAX=YABS
              IF(ZAIR(I).GT.ZMAX) ZMAX=ZAIR(I)
          10 CONTINUE
              VCOORD=MAX1(XMAX,YMAX)
C.
              XPLANE=30.
              YMIN=-VCOORD
              XMIN=-VCOORD
C.        SCALE DX TO DRAW RANGE CIRCLES AT 5 KM INTERVALS
              XLEN=VCOORD
              DX=5000.0
C.        CHECK FOR POSITION OF EYE SUCH THAT PLOT WILL ALWAYS BE LOOKING
C.        IN THE GENERAL DIRECTION OF FLIGHT STARTING FROM THE INITIAL
C.        FLIGHT PATH MILESTONE.
              VRAD=SQRT(XMAX**2 + YMAX**2) + 100.
              X1=XAIR(1)
              Y1=YAIR(1)
              THETA=ATAN4(y1,X1)
              XVIEW=VRAD*COS(THETA)
              YVIEW=VRAD*SIN(THETA)
C.        POSITION EYE POINT 2000M BEYOND VIEW POINT - THIS
C.        ORIENTS THE VIEWING DIRECTION
              EYERAD=VRAD+2000.
              XEYE=EYERAD*COS(THETA)
              YEYE=EYERAD*SIN(THETA)
C.
```

```
C.      ESTABLISH VIEW HEIGHT AT 1600 METERS WHICH EMPIRICALLY APPEARS
C.      TO BE OPTIMAL FOR PERSPECTIVE DISPLAY.
            ZVIEW=1600.0
            ZEYE=ZVIEW+1000.
C.
C.      FOR BEST PERSPECTIVE RESULTS, ENHANCE ALTITUDE SUCH THAT
C.      MAX ALTITUDE OCCURS AT THE VIEWING HEIGHT (1600M).  THIS
C.      PRESERVES THE PERSPECTIVE WHILE OPTIMIZING THE PLOT.
            BESTZ=1600.-ZMAX
            DO 20 J=1,NPOINT
            ZAIR(J)=ZAIR(J) + BESTZ
        20 CONTINUE
C.
C.      INITIATE PLOT
            CALL PLOTS(0,0,0)
            CALL SETMSG(3)
C
C.      GENERATE AND PLOT EARTH FIXED COORDINATE AXES
            CALL AXIS (0)
            CALL CGRID
C.      GENERATE AIRCRAFT FLIGHT PATH PASSING MAGNIFICATION FACTOR
            AMAG=15.
            CALL ACPLOT (NPOINT,L.,AMAG)
C
C.      PLOT TARGET AT ORIGIN WITH MAGNIFICATION FACTOR
            DO 30 I=1,3
            CG(I)=0.0
        30 CONTINUE
            IMAG=15.0
            CALL TARGET (CG,THEAD,TMAG)
C
C.      IF MISSILE ENVELOPE GREATER THAN INITIAL MILESTONE RANGE
            CALL LAUNCH (RNGMIS,MDISP)
C
C.      TERMINATE PLOT
        40 CALL PLOT(0.,0.,+999)
            RETURN
```

```
                END
      C
      C
                SUBROUTINE AXIS(NFLAG)
                DIMENSION X(5),Y(5),Z(5),XSCOPE(5),YSCOPE(5),CG(3),EULER(3)
                DIMENSION ZMAXX(5),ZMAXY(5),ZMINX(5),ZMINY(5),
               + XYPLNX(5),XYPLNY(5),X3(36),Y3(36),Z3(36),X4(36),Y4(36)
                DIMENSION WX(6),WY(6),XABS(5),YABS(5)
                COMMON/EVIEW/XVIEW,YVIEW,ZVIEW,XEYE,YEYE,ZEYE,XPLANE
                COMMON/GRID/XMIN, XMAX, YMIN,YMAX,DX,ZMAX,XORIG,YORIG
                COMMON/PSCALE/XVAL,YVAL,XV,YV,XSCALE,YSCALE
                COMMON/PARAM/MDISP,IAXIS,NPOINT
                COMMON/INPUT/SCALE?RNGMIS/THEAD
      C
      C.    SUBROUTINE PROJECTSX,Y,X COORDINATE AXES, INITIALIZES PLOTS,
      C.    AND GENERATES PLOTTING WINDOW SIZE.  IF NFLAG IS ZERO, THEN THE
      C.    AXES ARE NOT ACTUALLY PLOTTED BUT ARE COMPUTED FOR PURPOSE OF
      C.    DEFINING THE PLOTTING WINDOW.
      C.
      C.    SET UP AND PROJECT THE AXES
                DO 10 I=1,5
                X(I)=0.
                Y(I)=0.
                Z(I)=0.
                CONTINUE
                X(2)=XMAX
                Y(3)=YMAX
                Z(4)=ZMAX
                DO 20 I=1,3
                CG(I)=0.0
                EULER(I)=0.0
      20        CONTINUE
      CC.   MAKE PROJECTION OF COORDINATE AXES
                CALL CONPRO(CG,X,Y,Z,4,EULER,XSCOPE,YSCOPE)
      C
      C.    LOCATE LONGEST VECTOR PROJECTION AND SCALE A 9" PLOT
```

```
C.      LENGTH BY STEPPING AROUND A CIRCLE OF RADIUS=MAX UNPROJECTED
C.      LENGTH AT A 10 DEG INCREMENT
            RAD=MAX1(XMAX,YMAX)
            PI=3.14159
            DTHETA=10.*PI/180.
            THETA=0.
            DO 30 I=1,36
            X3(I)=RAD*COS(THETA)
            Y3(I)=RAD*SIN(THETA)
            Z3(I)=0.
            THETA=THETA+DTHETA
        30  CONTINUE
            CALL CONPRO(CG,X3,Y3,Z3,36,EULER,X4,Y4)
C.      LOCATE LONGEST PROJECTION VECTOR
            SFACT=0.
            DO 32 I=1,36
            DIST=SQRT(X4(I)**2 + Y4(I)**2)
            IF(DIST.GT.SFACT) SFACT=DIST
        32  CONTINUE
C.
            XSCALE=10.0*SCALE/SFACT
            YSCALE=XSCALE
            XORIG=XSCOPE(1)*XSCALE
            YORIG=-YSCOPE(1)*YSCALE
C.
C.      REORIGIN THE PLOT TO 11.0,10.0
            CALL PLOT (11.0,15.0,-3)
C
C.      IF NO PLOT DESIRED, LABEL PLOT AND RETURN
            IF(NFLAG.EQ.0) GO TO 100
C.      OTHERWISE PLOT THE AXES,TRANSLATE TO PLOT ORIGIN
            DO 40 I=2,4
            XABS(I)=XSCOPE(I)*XSCALE+XORIG
            YABS(I)=SCOPE(I)*YSCALE+YORIG
40          CONTINUE
            CALL PLOT(0.0,0.0,+3)
```

```
      CALL PLOT (XABS(3),YABS(3),+2)
      CALL WHERE (YLABX,YLABY,DFACT)
      CALL PLOT(0.0,0.0,+3)
      CALL PLOT(XABS(4),YABS(4),+2)
      CALL WHERE (ZLABX,ZLABY,DFACT)
C
C.    LABEL AXES
      CALL SETMSG(3)
      CALL SYMBOL(XLABX,XLABY,.2,'X',0.,1)
      CALL SYMBOL(YLABX,YLABY,.2,'Y',0.,1)
      CALL SYMBOL(ZLABX,ZLABY,.2,'Z',0.,1)
C.
C.    LABEL THE PLOT AND RETURN
  100 CALL NEWPEN(3)
      CALL SYMBOL(-2.0,5.5,.5,'SEA PLOT',0.,8)
      CALL NEWPEN(1)
      CALL SYMBOL(-2.4,4.5,.1,'PERSPECTIVE VIEW COORD:',
     +0.,24)
      CALL NUMBER(.5,4.5,.1,XVIEW,0.,0)
      CALL NUMBER(1.25,4.5,.1,YVIEW,0.,0)
      CALL NUMBER(2.,4.5,.1,ZVIEW,0.,0)
C.    LABEL THE RANGE RINGS
      CALL SYMBOL(-2.0,4.25,.1,'EACH RANGE RING=',0.,17)
      CALL NUMBER(0.,4.25,.1,DX,0.0,0)
      CALL SYMBOL(.75,4.25,.1,'METERS',0.,6)
C
C.    LABEL THE LAUNCH ENVELOPE
      CALL SYMBOL(-2.0,4.0,.1,'LAUNCH ENVELOPE AT',0.,18)
      CALL NUMBER(0.,4.0,.1,RNGMIS,0.,0)
      CALL SYMBOL(.75,4.0,.1,'METERS',0.,6)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE DRAW(X,Y,N,ITYPE)
      DIMENSION X(N),Y(N),XPLOT(200),YPLOT(200)
      COMMON/PSCALE/XVAL,YVAL,XV,YV,XSCALE,YSCALE
      COMMON/GRID/XMIN,XMAX,YMIN,YMAX,DX,ZMAX,XORIG,YORIG
C
C.    SUBROUTINE CAUSES A PLOT MOVE TO THE 1ST COORDINATE PAIR
C.    IN X,Y THEN A VECTOR DRAW TO THE REMAINING POINTS.  INPUT ARRAY
C.    VALUES ARE SCALED TO THE PLOTTER UNITS WITHOUT DESTROYING THE
C.    ORIGINAL ARRAY VALUES.  VARIABLE ITYPE INDICATES WHETHER VECTORS
C.    ARE TO BE DRAWN AS DOTS OR SOLID LINES (1=LINES,0=DOTS)
C
C.    SCALE TO PLOTTER UNITS AND TRANSLATE TO PLOT ORIGIN
      DO 10 I=1,N
      XPLOT(I)=X(I)*XSCALE +XORIG
      YPLOT(I)=Y(I)*YSCALE +YORIG
10    CONTINUE
C
C.    MOVE TO FIRST VALUE
      CALL PLOT(XPLOT(1),YPLOT(1),+3)
C
C.    CHECK TYPE OF VECTOR TO DRAW
      IF(ITYPE.EQ.L)GO TO 30
C.    DRAW PLOT USING DOTS
      DO 20 I=2,N
      CALL PLOT(XPLOT(I),YPLOT(I),+3)
      CALL WHERE(XDOT,YDOT,DFACT)
      CALL SYMBOL(XDOT,YDOT,.1,'.',0.,0)
20    CONTINUE
      GO TO 100
C
C.    DRAW PLOT USING LINES
   30 DO 40 I=2,N
      CALL PLOT(XPLOT(I),YPLOT(I),+2)
40    CONTINUE
C
100   RETURN
      END
```

```
C
C
      SUBROUTINE VDASH(X1,Y1,X2,Y2)
      COMMON/PSCALE/XVAL,YVAL,XV,YV,XSCALE,YSCALE
      COMMON/GRID/XMIN,XMAX,YMIN,YMAX,DX,ZMAX,XORIG,YORIG
C.
C.    SUBROUTINE USES VERSATEC SOFTWARE TO DRAW A DASHED LINE
C.    BETWEEN 2 POINTS PASSED AS ARGUMENTS.  ONLY VERTICAL
C.    LINES ARE DRAWN.  LINE CONSISTS OF 20 DOTS DRAWN BETWEEN POINTS.
C
      CALL NEWPEN(1)
      X=X1*XSCALE+XORIG
      Y=Y1*YSCALE+YORIG
      DY=(Y2-Y1)*YSCALE/20.0
      DO 10 I=1,20
      CALL PLOT(X,Y,+3)
      CALL WHERE(XDOT,YDOT,DFACT)
      CALL SYMBOL(XDOT,YDOT,.05,'.',0.k0)
      Y=Y+DY
   10 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE MULT(D,X,Y,Z,XNEW,YNEW,ZNEW)
      DIMENSION D(3,3)
C
C.    SUBROUTINE PERFORMS MATRIX MULTIPLICATION OF MATRIX D ON COLUMN
C.    VECTOR GIVEN BY X,Y,Z
C
      XNEW=D(1,1)*X +D(1,2)*Y +D(1,3)*Z
      YNEW=D(2,1)*X +D(2,2)*Y +D(2,3)*Z
      ZNEW=D(3,1)*X +D(3,2)*Y +D(3,3)*Z
      RETURN
      END
C
C
```

```
      SUBROUTINE DIRCOS(PSI,THETA,PHI,D,TRANSP)
      DIMENSION D(3,3)
C.    SUBROUTINE COMPUTES DIRECTION COSINE MATRIX D
C
      SPSI=SIN(PSI)
      CPSI=COS(PSI)
      STHET=-SIN(THETA)
      CTHET=COS(THETA)
      SPHI=SIN(PHI)
      CPHI=COS(PHI)
      D(1,1)=CPSI*CTHET
      D(1,2)=SPSI*CTHET
      D(1,3)=-STHET
      D(2,1)=CPSI*STHET*SPHI-SPSI*CPHI
      D(2,2)=SPSI*STHET*SPHI+CPSI*CPHI
      D(2,3)=CTHET*SPHI
      D(3,1)=CPSI*STHET*CPHI+SPSI*SPHI
      D(3,2)=SPSI*STHET*CPHI-CPSI*SPHI
      D(3,3)=CTHET*CPHI
C
C.    TRANSPOSE OF D MATRIX:
      IF(TRANSP.EQ.0) RETURN
      DO 10 I1=1,3
      DO 10 I2=I1,3
      TEMP=D(I1,I2)
      D(I1,I2)=D(I2,I1)
      D(I2,I1)=TEMP
10    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE CONPRO(CG,XP,YP,ZP,NPOINT,EULER,YSCOPE,ZSCOPE)
      DIMENSION CG(3),XP(NPOINT),YP(NPOINT),ZP(NPOINT),EULER(3)
      DIMENSION YSCOPE(NPOINT),ZSCOPE(NPOINT),A(3,3),C(3,3)
      COMMON/EVIEW/XVIEW,YVIEW,ZVIEW,XEYE,YEYE,ZEYE,XPLANE
```

```
C.        SUBROUTINE MAKES A CONIC PROJECTION OF POINTS IN A 3 SPACE PASSED
C.        IN XP,YP,ZP.  PROJECTION BASED ON VIEWING COORDINATES, CENTER OF
C.        GRAVITY, AND EULER ANGLES (ROLL,PITCH,HEADING).  XPLANE IS THE
C.        DISTANCE FROM VIEWER TO THE PROJECTION PLANE.
C
              PI=4.*ATAN(1.)
              PSI=EULER(1)*PI/180.
              THETA=EULER(2)*PI/180.
              PHI=EULER(3)*PI/180.
C
C.        GENERATE DIRECTION COSINE MATRIX FOR TRANSFORM FROM BODY TO
C.        EARTH AXES
              CALL DIRCOS(PSI,THETA,PHI,C,1)
C
C.        GENERATE DIRECTION COSINE MATRIX FOR TRANSFORM FROM BODY TO
C.        EARTH AXES
              CALL DIRCOS(PSI,THETA,PHI,C,1)
C
C.        ROTATE EARTH COORDINATES TO EYE-VIEW SYSTEM
              XARC=XVIEW-XEYE
              YARC=YVIEW-YEYE
              PSIEV=ATAN4(YARC,XARC)
              R2D=SQRT(XARC**2 + YARC**2)
              YARC=ZVIEW-ZEYE
              THETEV=ATAN4(YARC,R2D)
              CALL DIRCOS(PSIEV,THETEV,0.,A,0)
C
              DO 30 I=1,NPOINT
C
C.        GENERATE BODY COORDINATES IN INERTIAL COORDINATE SYSTEM:XPR,YPR, ZPR
              CALL MULT(C,XP(I),YP(I),ZP(I),XPR,YPR,ZPR)
C
C.        TRANSLATE TO ORIGIN OF EYE-VIEW SYSTEM
              XPR=XPR+CG(1)-XEYE
              YPR=YPR+CG(2)-YEYE
              ZPR=ZPR+CG(3)-ZEYE
```

```
C
C.      ROTATE TO EYE-VIEW SYSTEM, GENERATE BODY COORDINATES IN EYE-VIEW SYS
           CALL MULT(A,XPR,YPR,ZPR,XE,YE,ZE)
C
C.      CONVERT TO PROJECTION PLANE COORDINATES (PLOTTER COORDINATES)
           YSCOPE(I)=-YE*XPLANE/XE
           ZSCOPE(I)=ZE*XPLANE/XE
C
        30 CONTINUE
           RETURN
           END
C
C
           SUBROUTINE CGRID
           DIMENSION X(181),Y(181),Z(181),YSCOPE(181),ZSCOPE(181)
           DIMENSION CG(3),EULER(3)
           COMMON/EVIEW/XVIEW,YVIEW,ZVIEW,XEYE,YEYE,ZEYE,XPLANE
           COMMON/GRID/XMIN,XMAX,YMIN,YMAX,DX,ZMAX,XORIG,YORIG
           COMMON/PSCALE/XVAL,YVAL,XV,YV,XSCALE,YSCALE
C
C.      SUBROUTINE PLOTS PROJECTED RANGE CIRCLES (AS DOTS) CENTERED
C.      AT THE PROJECTION ORIGIN.  NUMBER OF POINTS IN EACH CIRCLE
C.      INCREASES FROM 20 TO 180 AS DETERMINED BY THE RADIUS OF THE
C.      CIRCLE.
C
C.      SET UP RANGE INCREMENT AND LOCATE THE ORIGIN
           PI=4.0*ATAN(1.)
           RADINC=XMAX/DX
           INC=IFIX(RADINC)
           RADIUS=0.0
           X(1)=0.0
           Y(1)=0.0
           DO 5 I=1,181
           Z(I)=0.0
5          CONTINUE
```

```
            DO 10 I=1,3
            CG(I)=0.0
            EULER(I)=0.0
10          CONTINUE
C
C       DRAW RANGE RINGS IN FINER LINES
            CALL NEWPEN(1)
C.      DEFINE AND PROJECT THE RANGE CIRCLES PLOTTING 181 VECTORS
C.      PER CIRCLE
            MAXINC=INC+1
            DTHETA=2.0*PI/180.
            DO 100 I=1,MAXINC
            RADIUS=RADIUS+DX
            THETA=0.0
C
            DO 30 J=1,181
            X(J)=RADIUS*COS(THETA)
            Y(J)=RADIUS*SIN(THETA)
            THETA=THETA+DTHETA
        30  CONTINUE
            CALL CONPRO(CG,X,Y,Z,181,EULER,YSCOPE,ZSCOPE)
            CALL DRAW(YSCOPE,ZSCOPE,181,1)
C
100         CONTINUE
C
C       RESTORE PEN
            CALL NEWPEN(1)
C
            RETURN
            END
C
C
            SUBROUTINE ACPLOT(N,VLINE,AMAG)
            COMMON/AIR/XAIR,YAIR,ZAIR,ACROLL,ACPTCH,ACHEAD
            DIMENSION XAIR(200),YAIR(200),ZAIR(200)
```

```
      DIMENSION X(200),Y(200),Z(200)
      DIMENSION ACROSS(200),ACPTCH(200),ACHEAD(200)
      DIMENSION XPLOTA(200),YPLOTA(200),CG(3),EULER(3)
      DIMENSION XPLOTG(200),YPLOTG(200)
      COMMON/EVIEW/XVIEW,YVIEW,ZVIEW,XEYE,YEYE,ZEYE,XPLANE
      COMMON/GRID/XMIN,XMAX,YMIN,YMAX,DX,ZMAN,XORIG,YORIG
      COMMON/PSCALE?XVAL,YVAL,XV,YV,XSCALE,YSCALE
C
C.    SUBROUTINE PLOTS UP TO 200 AIRCRAFT MILESTONES SHOWING BOTH AIR
C.    TRACK AND GROUND PROJECTION.  EVERY 4TH MILESTONE, THE AIRCRAFT
C.    BODY IS DRAWN SHOWING AIRCRAFT ATTITUDE.  INPUT ARRAYS ARE PRESERVED
C.    DURING COMPUTATION.
C
C=========PLOT SURFACE TRACK=========
C
C.    SET AIRCRAFT ALTITUDE TO 0 AND PLOT.  SINCE NO BODY COORDINATES
C.    ARE NEEDED, SET CENTER OF GRAVITY AND EULER ANGLES TO 0
      DO 10 I=1,N
      X(I)=XAIR(I)
      Y(I)=YAIR(I)
      Z(I)=0.0
   10 CONTINUE
      DO 20 I=1,3
      CG(I)=0.0
      EULER(I)=0.0
   20 CONTINUE
      MAKE PROJECTION OF GROUND TRACK - STORE PLOT COORDINATES IN
C.    XPLOTG AND YPLOTG.
      CALL CONPRO(CG,X,Y,Z,N,EULER,XPLOTG,YPLOTG)
C.    PLOT GROUND TRACK
      CALL NEWPEN(5)
      CALL DRAW(XPLOTG,YPLOTG,N,0)
C
C=========================================
C
C.============PLOT AIR TRACK=================
```

```
C
C.      DEFINE MAGNIFICATION VALUE FOR DISPLAY OF A/C BODY
            CALL NEWPEN(2)
C.      SET UP MILESTONE COUNTER FOR DISPLAY OF A/C BODY
            DO 50 I=1,N
            X(I)=XAIR(I)
            Y(I)=YAIR(I)
            Z(I)=ZAIR(I)
        50  CONTINUE
C.      RESET CENTER OF GRAVITY AND EULER ANGLES, THEN PROJECT AND PLOT
C.      FLIGHT PATH IN DOTS.
            DO 60 J=1,3
            EULER(J)=0.0
            CG(J)=0.0
        60  CONTINUE
            CALL CONPRO(CG,X,Y,Z,N,EULER,XPLOTA,YPLOTA)
            CALL DRAW(XPLOTA,YPLOTA,N,1)
C
C.      CONNECT GROUND AND SURFACE TRACKS WITH DASHED LINE
C.      TO ENHANCE PERSPECTIVE
            IF(VLINE.NE.1.) GO TO 72
            DO 70 I=1,N
            CALL VDASH(XPLOTA(I),YPLOTA(I),XPLOTG(I),YPLOTG(I))
        70  CONTINUE
C.
C.      DRAW PERSPECTIVE A/C BODY EVERY 2 KM OF GROUND TRACK
C.      BODY TO BE PLOTTED ALONG THE AIR TRACK
        72 CALL NEWPEN(3)
C.      PLOT INITIAL BODY POSITION
            XCG=XAIR(1)
            YCG=YAIR(1)
            ZCG=ZAIR(1)
            EULER(1)=ACHEAD(1)
            EULER(2)=ACPTCH(1)
            EULER(3)=ACROSS(1)
            CG(1)=XCG
```

```
      CG(2)=YCG
      CG(3)=ZCG
      CALL ACRAFT(EULER,CG,AMAG)
C.    CHECK DISTANCE TRAVELED OVER GROUND AND PLOT AT 2 KM STEPS
      DO 80 I=2,N
      X2=XAIR(I)
      Y2=YAIR(I)
      Z2=ZAIR(I)
      DIST=SQRT((X2-XCG)**2 + (Y2-YCG)**2)
      IF(DIST.LT.2000.) GO TO 80
      XCG=X2
      YCG=Y2
      ZCG=Z2
      CG(1)=XCG
      CG(2)=YCG
      CG(3)=ZCG
      EULER(1)=ACHEAD(I)
      EULER(2)=ACPTCH(I)
      EULER(3)=ACROSS(I)
      CALL NEWPEN(5)
      CALL ACRAFT(EULER,CG,AMAG)
   80 CONTINUE
C
C.    RESTORE PEN
      CALL NEWPEN(1)
C
      RETURN
      END
C
C
      SUBROUTINE ACRAFT(EULER,CG,AMAG)
      DIMENSION CG(3),EULER(3)
      DIMENSION XAC(8),YAC(8),ZAC(8),X(8),Y(8),Z(8),XPLOT(8),YPLOT(8)
C.
C.    PREDEFINED AIRCRAFT BODY COORDINATES:
```

```
          DATA XAC/25.,-15.,-15.,25.,0.,-15.,-15.,0./
          DATA YAC/0.,-10.,10.,5*0./
          DATA ZAC/5*0.,5.,0.,0./
C.
C.      SUBROUTINE PLOTS PREDEFINED AIRCRAFT BODY COORDINATES, POSITIONING
C.      THE BODY IN 3 SPACE AS SPECIFIED BY EULER ANGLES AND CENTER OF GRAVI
C
C.      APPLY MAGNIFICATION FACTOR TO BODY COORDINATES
          DO 5 I=1,8
          X(I)=XAC(I)*AMAG
          Y(I)=YAC(I)*AMAG
          Z(I)=ZAC(I)*AMAG
        5 CONTINUE
C
C.      MAKE THE PROJECTION
          CALL CONPRO(CG,X,Y,Z,8,EULER,XPLOT,YPLOT)
C.      DRAW THE AIRCRAFT
          CALL DRAW(XPLOT,YPLOT,8,1)
C
C.
          RETURN
          END
C
C
          SUBROUTINE TARGET(CG,THEAD,TMAG)
          DIMENSION XTGT(26),YTGT(26),ZTGT(26),XSCOPE(26),YSCOPE(26)
          DIMENSION X(26),Y(26),Z(26),CG(3),EULER(3),IPAT(5)
          COMMON/EVIEW/XVIEW,YVIEW,ZVIEW,XEYE,YEYE,ZEYE,XPLANE
          COMMON/GRID/XMIN,XMAX,YMIN,YMAX,DX,ZMAX,XORIG,YORIG
          COMMON/PSCALE/XVAL,YAVL,XV,YV,XSCALE,YSCALE

C.      SUBROUTINE TARGET PROJECTS AND PLOTS A GENERIC SHIP IMAGE CENTERED
C.      AT COORDINATES PASSED IN CG ARRAY, WITH HEADING PASSED IN THEAD.
C.      IMAGE IS MAGNIFIED BY A FACTOR OF TMAG.  TARGET BODY COORDINATES ARE
C.      PREDEFINED IN THIS ROUTINE.
```

```
C.      AS WRITTEN, THIS ROUTINE DRAWS ONLY THE LOWER PLANE OF THE HULL.
C.      THIS IS TO RELIEVE A CROWDED PLOT.  HOWEVER, IF A HIDDEN LINE
C.      ROUTINE IS INCLUDED WHICH WILL CLEAN UP THE PLOT, THE REST
C.      OF THE BODY COORDINATES ARE PROVIDED IN THE DATA STATEMENTS.
C
C.      BODY COORDINATES (IN METERS);
            DATA XTGT/75.,51.,-75.,-75.,56.,75.,75.,51.,-75.,-75.,51.,75.,
            +15.,-15.,-15.,3*15.,2*-15.,2*15.,4*0./
            DATA YTGT/0.,2*-12.5,2*12.5,2*0.,2*-12.5,2*12.5,0.,2*-4.5,
            +2*4.5,3*-4.5,2*4.5,-4.5,2*0.,12.5,-12.5/
            DATA ZTGT/6*0.,11*5.,6*15.,30.,2*25./
            DATA IPAT/ZFFFF,Z1111,Z1111,Z1111,Z1111/
C.
C.      DRAW TARGET WITH BOLDER LINES
            CALL NEWPEN(4)
C.      SET UP FULER ANGLES FOR PROJECTION
            EULER(1)=THEAD
            EULER(2)=0.
            EULER(3)=0.
C.      PROJECT, MAGNIFY AND DRAW THE HULL (1ST 12 DATA POINTS)
            DO 10 I=1,6
            X(I)=XTGT(I)*TMAG
            Y(I)=YTGT(I)*TMAG
            Z(I)=ZTGT(I)*TMAG
        10  CONTINUE
            CALL CONPRO(CG,X,Y,Z,6,EULER,XSCOPE,YSCOPE)
            CALL DRAW(XSCOPE,YSCOPE,6,1)
C
C.      SHADE IN THE TARGET WITH PATTERN DEFINED IN IPAT
C.      SHIFT DISPLAY COORDINATES TO CONFORM TO DISPLAY SCALE
C.      AND ORIGIN
            DO 20 I=1,6
            XSCOPE(I)=XSCOPE(I)*XSCALE+XORIG
            YSCOPE(I)=YSCOPE(I)*YSCALE+YORIG
        20  CONTINUE
```

```
      CALL TONE(0.,0.,IPAT,-5)
      CALL TONE(XSCOPE,YSCOPE,6,+1)
C
C.    RESET PEN
      CALL NEWPEN(1)
C
      RETURN
      END
C
C
      SUBROUTINE LAUNCH(RANGE,IDISPL)
      DIMENSION X(181),Y(181),Z(181)XPLOT(181),YPLOT(181)
      DIMENSION CG(3),EULER(3)
      COMMON/EVIEW/XVIEW,YVIEW,ZVIEW,XEYE,YEYE,ZEYE,XPLANE
      COMMON/GRID/XMIN,XMAX,YMIN,YMAX,DX,ZMAX,XORIG,YORIG
      COMMON/PSCALE/XVAL,YVAL,XV,YV,XSCALE,YSCALE
C
C.    SUBROUTINE LAUNCH DRAWS SPHERICAL LAUNCH ENVELOPE CORRESPONDING
C.    TO MISSILE SLANT RANGE PASSED AS A PARAMETER.  USES STANDARD
C.    CYLINDRICAL TO CARTESIAN TRANSFORM.  IF IDISPL=1, PLOT MISSILE
C.    ENVELOPE AS A BOLD RANGE CIRCLE.
C
C.
C.    SET UP
      DO 5 I=1,3
      CG(I)=0.0
      EULER(I)=0.0
    5 CONTINUE
      PI=4.0*ATAN(1.)
      DZ=2.0*ZMAX/10.
      CALL NEWPEN(1)
C.    ANGLE INCREMENTS:  DTHETA=HORIZONTAL
      Z1=0.0
      DTHETA=2.0*PI/180.
C.    CHECK FOR TYPE OF DISPLAY
      IF(IDISPL.EQ.1) GO TO 65
```

```
C
C.      ITERATE THROUGH THE VERTICAL ANGLE MAKING TRANSFORM
            DO 60 I=1,10
C.      CHECK TO ENSURE THAT PLOT DOES NOT EXCEED MAX PLOT ELEVATION
            Z1=Z1+DZ
C.      DRAW HORIZONTAL RANGE RINGS
            THETA=0.
            DO 40 J=1,180
            X(J)=RANGE*COS(THETA)
            Y(J)=RANGE*SIN(THETA)
            Z(J)=Z1
            THETA=THETA=DTHETA
        40  CONTINUE
        PROJECT AND DRAW RANGE RING USING DOTS
            CALL CONPRO(CG,X,Y,Z,180,EULER,XPLOT,YPLOT)
            CALL DRAW(XPLOT,YPLOT,180,0)
C.      INCREMENT VERTICAL ANGLE AND REPEAT
        50  PHI=PHI+DELPHI
        60  CONTINUE
            GO TO 80
C
C.      PLOT RANGE CIRCLE
        65  THETA=0.
            CALL NEWPEN(5)
            DO 70 I=1,181
            X(I)=RANGE*COS(THETA)
            Y(I)=RANGE*SIN(THETA)
            Z(I)=0.
            THETA=THETA+DTHETA
        70  CONTINUE
            CALL CONPRO(CG,X,Y,Z,181,EULER,XPLOT,YPLOT)
            CALL DRAW(XPLOT,YPLOT,181,1)
C
        80  RETURN
            END
```

```
C
C
            FUNCTION ATAN4(Y,X)
C
C.     FUNCTION SOLVES ARCTANGENT FOR AN ANGLE BETWEEN 0 AND 2PI RADIANS
C
            IF(X.NE.0.) GO TO 50
            IF (Y) 10,30,20
       10   ATAN4=4.712388981
            GO TO 100
X
       20   ATAN4=1.570796327
            GO TO 100
C
       30   ATAN4=0.
            GO TO 100
C
       50   ATAN4=ATAN(Y/X)
            IF(Y.LT.0.) GO TO 70
            IF(X.GE.0.) GO TO 100
C
       60   ATAN4=ATAN4 + 3.141592654
            GO TO 100
C
       70   IF(X.LT.0.) GO TO 60
            ATAN4=ATAN4 + 6.283185308
C
       100  RETURN
            END
/*
//GO.PLOTPARM DD *
 &PLOT SCALE=0.5 &END
/*
/GO.SYSIN DD *
C. INPUT DATA FOLLOWS HERE
```

APPENDIX B

LISTING OF CONIC PROJECTION ROUTINES

```
          SUBROUTINE CONPRO(CG,XP,YP,ZP,NPOINT,EULER,YSCOPE,ZSCOPE)
          DIMENSION CG(3),XP(NPOINT),YP(NPOINT),ZP(NPOINT),EULER(3)
          DIMENSION YSCOPE(NPOINT),ZSCOPE(NPOINT),A(3,3),C(3,3)
          COMMON/EVIEW/XVIEW,YVIEW,ZVIEW,XEYE,YEYE,ZEYE,XPLANE
C
C.      SUBROUTINE MAKES A CONIC PROJECTION OF POINTS IN 3 SPACE PASSED
C.      IN XP,YP,ZP.  PROJECTION BASED ON VIEWING COORDINATES, CENTER OF
C.      GRAVITY, AND EULER ANGLES (ROLL, PITCH,HEADING).  XPLANE IS THE
C.      DISTANCE FROM VIEWER TO THE PROJECTION PLANE.
C
          PI=4.*ATAN(1.)
          PSI=EULER(1)*PI/180.
          THETA=EULER(2)*PI/180.
          PHI=EULER(3)*PI/180.
C
C.      GENERATE DIRECTION COSINE MATRIX FOR TRANSFORM FROM BODY TO
C.      EARTH AXES
          CALL DIRCOS(PSI,THETA,PHI,C,1)
C
C.      ROTATE EARTH COORDINATES TO EYE-VIEW SYSTEM
          XARC=XVIEW-XEYE
          YARC=YVIEW-YEYE
          PSIEV=ATAN4(YARC,XARC)
          R2D=SQRT(XARC**2 + YARC**2)
          YARC=ZVIEW-ZEYE
          THETEV=ATAN4(YARC,R2D)
          CALL DIRCOS(PSIEV,THETEV,0.,A,0)
C
```

```
      DO 30 I=1,NPOINT
C
C.    GENERATE BODY COORDINATES IN INERTIAL COORDINATE SYSTEM:XPR,YPR,ZPR
      CALL MULT(C,XP(I),YP(I),ZP(I),XPR,YPR,ZPR)
C
C.    TRANSLATE TO ORIGIN OF EYE-VIEW SYSTEM
      XPR=XPR+CG(1)-XEYE
      YPR=YPR+CG(2)-YEYE
      ZPR=ZPR+CG(3)-Zeye
C
C.    ROTATE TO EYE-VIEW SYSTEM, GENERATE BODY COORDINATES IN EYE-VIEW SYS
      CALL MULT(A,XPR,YPR,ZPR,XE,YE,ZE)
C
C.    CONVERT TO PROJECTION PLANE COORDINATES (PLOTTER COORDINATES)
      YSCOPE(I)=-YE*XPLANE/XE
      ZSCOPE(I)=ZE*XPLANE/XE
C
   30 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE DIRCOS(PSI,THETA,PHI,D,TRANSP)
      DIMENSION D(3,3)
C
      SUBROUTINE COMPUTES DIRECTION COSINE MATRIX D

      SPSI=SIN(PSI)
      CPSI=COS(PSI)
      STHET=-SIN(THETA)
      CTHET=COS(THETA)
      SPHI=SIN(PHI)
      CPHI=COS(PHI)
```

```
      D(1,1)=CPSI*CTHET
      D(1,2)=SPSI*CTHET
      D(1,3)=-STHET
      D(2,1)=CPSI*STHET*SPHI-SPSI*CPHI
      D(2,2)=SPSI*STHET*SPHI+CPSI*CPHI
      D(2,3)=CTHET*SPHI
      D(3,1)=CPSI*STHET*CPHI+SPSI*SPHI
      D(3,2)=SPSI*STHET*CPHI-CPSI*SPHI
      D(3,3)=CTHET*CPHI
C
C.    TRANSPOSE OF D MATRIX:
      IF(TRANSP.EQ.0) RETURN
      DO 10 I1=1,3
      DO 10 I2=I1,3
      TEMP=D(I1,I2)
      D(I1,I2)=D(I2,I1)
      D(I2,I1)=TEMP
10    CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE MULT(D,X,Y,Z,XNEW,YNEW,ZNEW)
      DIMENSION D(3,3)
C
C     SUBROUTINE PERFORMS MATRIX MULTIPLICATION OF MATRIX D ON COLUMN
      VECTOR GIVEN BY X,Y,Z
C
      XNEW=D(1,1)*X +D(1,2)*Y +D(1,3)*Z
      YNEW=D(2,1)*X +D(2,2)*Y +D(2,3)*Z
      ZNEW=D(3,1)*X +D(#,2)*Y +D(#,3)*Z
      RETURN
      END
C
```

APPENDIX C

GRAPHICS SOFTWARE PLOT 80

USING THE TEKTRONIX 4081 IN THE PROGRAM MODE

I. INTRODUCTION

This appendix discusses the Tektronix graphics software package PLOT 80 and its use on the Tektronix 4081 graphics computer. Because Reference 6 provides an adequate discussion of the Tektronix 4081 equipment and system operation, computer light-off and operation will not be reiterated here. Instead, discussion will be focused on the graphics language PLOT 80 and its uses in the Tektronix 4081 program mode.

PLOT 80 is a particularly powerful graphics tool consisting of 374 different subroutines covering a wide range of graphics concepts including bit stream manipulation, vector drawing, test input and output, and picture display. These subroutines are discussed in detail in the PLOT 80 user's manual, Reference 7. Because PLOT 80 represents a complex, sophisticated software system, Reference 7 should be reviewed by a prospective user prior to any attempt at programming of the Tektronix 4081. This appendix will discuss some of the more basic concepts employed by PLOT 80 and provide the programmer some useful background information which is intended to complement the information in Reference 7.

PLOT 80 will be discussed here in the context of the 4081 program mode of operation as opposed to the host mode

discussed in Reference 6. In the program mode, all computations are accomplished using the hardware organic to the 4081 system itself; unlike the host mode in which a host computer, such as the IBM 3033 or PDP-11, performs all calculations and uses the 4081 for graphics display only. The word size of the 4081 does inhibit many mathematical computations, such as the transforms used in SEA PLOT, which result in underflows and hence limits to some extent the program mode applications of the 4081. However, this mode does provide an excellent tool for investigating graphics structures such as the target and aircraft body representations used in SEA PLOT. Section II of this appendix discusses some of the important basic concepts of PLOT 80. Section III will describe the procedures for file creation, compiling, linking, and execution on the Tektronix 4081.

II. PLOT 80

The 374 distributed graphics support subroutines (DGSS) in PLOT 80 are FORTRAN callable routines that permit the user to:

(1) Perform sophisticated graphics in integer or floating point format. For example, routines that perform graphic transformations or create polygons, circles, and splines are included in the software package.

(2) Perform graphic input (GIN) in windowspace or viewspace coordinates.

(3) Perform file manipulation.

(4) Communicate with the host computer.

(5) Perform assembly language operations including bit and character string manipulation, primitive I/O, and dynamic memory allocation.

Before performing any DGSS graphics, a program must include the two following steps:

CALL BFDEV (logical unit, device).

CALL GONEW (logical unit).

The first routine assigns a logical unit (to be defined in following paragraphs) to a graphic output device; i.e., the display controller. The second routine creates a data structure called the graphics control block (GCB) for the specified logical unit. By means of the logical unit,

therefore, a GCB is linked with a graphic output device as shown in Figure A-1.

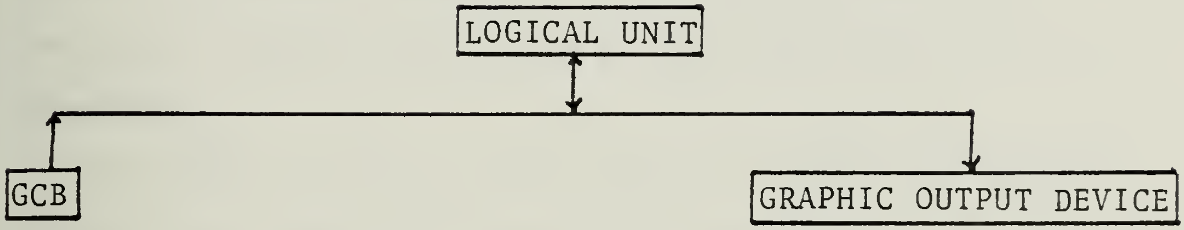

Figure A-1

For example, if the programmer decides that the display controller is to be assigned to logical unit 0, he could create a GCB as follows:

```
CALL BFDEV (0,'DC:')
CALL GONEW (0)
```

He is now ready to create the graphics control block for his particular graphics needs.

The graphic control block contains a sequence of parameters that determine exactly where on the surface of the graphic output device the graphics information defined in a program is to be displayed. The GCB is essential to all DGSS graphics. Attempting to output graphics information to a device without first creating a GCB causes an error condition to be signaled which will terminate execution.

No DGSS graphics programmer can entirely ignore the concept of a GCB. The GONEW routine, however, allows a programmer to get directly to the problem of the graphics application without spending much time and space on creating a GCB and defining each and every one of its parameters.

The GONEW routine automatically allocates the memory required for a GCB and initializes the contents to a set of default values. The programmer does not need to know where in memory the GCB is located or how much space it occupies. Table 14-1 of Reference 7 lists the GCB parameters and the default settings.

Data transfers under the graphics operating system are always performed through a logical unit assigned to a device. The programmer communicates with the logical unit rather than the device itself. This process greatly simplifies the I/O operation. The programmer does not need to do "housekeeping" tasks, such as addressing the device and checking the status of the device to see if it is busy or ready to send and receive data. All housekeeping is performed by the proper device driver which is called into action when a device is assigned to a logical unit. There are 16 logical units, numbered 0 to 15. Initially, all units are assigned to the keyboard unit until a BFDEV call is made. Any logical unit may be assigned to any device. Graphic output devices are:

(1) display controller
(2) plotter
(3) a file

Another important concept available to the programmer is that of a picture segment. Using this concept, the overall plotted output is divided into collections of vectors called picture segments. Each segment is numbered and can be specified independently of the other segments. The programmer can therefore manipulate each segment independently of the others. (Recall that SEA PLOT was organized along these lines with the coordinate axes, surface grid, launch envelope, and flight path each being a separate picture segment capable of individual manipulation.)

Picture segments are also sometimes called refresh objects, which refers to how the segment is displayed on the screen. Segments can be displayed in either refresh or storage. When a segment is drawn in storage, the segment is drawn once on the screen and remains visible until the screen is erased. A segment displayed in refresh, however, is not "stored" on the screen. The segment must be continually drawn and redrawn on the screen to remain visible (a function automatically carried out and not a programmer responsibility).

Refresh graphics allow the modification and movement of segments without erasing the screen. The effect is similar to animation in that a picture is drawn at one point on the screen, then redrawn again at another, then again at

another, and so on. The picture seems to be moving across the screen.

There are three steps in creating a picture segment:

(1) Initialize the segment assigning it a number (up to 255).

(2) Specify vector attributes (coordinates, intensity, etc.).

(3) Terminate segment creation.

These are accomplished by the OPEN and CLOSE subroutines.

To illustrate the use of the GCB and picture segments, a possible animation routine for SEA PLOT would be as follows:

```
        CALL BFDEV (0,'DC:')
        CALL GONEW(0)
C.  DRAW COORDINATE AXES (PICTURE SEGMENT 1)
        CALL OPEN (1)
        CALL AXIS (1)
        CALL CLOSE (1)
C.  PLOT AXES IN STORAGE
        CALL FIX (1)
C.  DRAW RANGE RINGS (PICTURE SEGMENT 2)
        CALL OPEN (2)
        CALL CGRID
        CALL CLOSE (2)
C.  PLOT RANGE RINGS IN STORAGE
        CALL FIX (2)
```

```
C.   DRAW AIRCRAFT (SEGMENT 3) IN REFRESH MOVING FROM
C.   MILESTONE TO MILESTONE
         DO 10 I=1,NPOINT
         READ X,Y,Z,ROLL,PITCH,HEAD
         CALL OPEN (3)
         CALL ACRAFT(Z,Y,Z,ROLL,PITCH,HEAD)
         CALL CLOSE (3)
C.   PLOT SEGMENT 3 IN REFRESH
         CALL POST (3)
C.   TIME DELAY CORRESPONDING TO AIRCRAFT VELOCITY
         CALL TDELAY
C.   ERASE PREVIOUS BODY PLOT
         CALL UNPOST (3)
C.   ITERATE THROUGH REMAINING MILESTONES
     10  CONTINUE
         STOP
         END
```

III. FILE CREATION AND EXECUTION

This final section discusses Tektronix 4081 program mode procedures to create, compile, link-edit, and execute a program. Unlike the job control language options of the NPS IBM 3033 main frame computer, compiling, linking, and executing jobs on the 4081 are entirely distinct operations which must be individually performed by the operator.

A. FILE CREATION

Creating a local file on the 4081 is accomplished by the command "EDIT filename. file type." This operation is essentially the same, using the same functions, as the EDIT system used on the main frame. When creating a multi-structured file, one must first create a library file. This library file will then contain each of the individual subroutines comprising the overall program. Moreover, each subroutine must be individually compiled.

Using SEA PLOT as an example, one would create a library file named SEAPLT containing all of the applicable subroutines as follows:

FORMAT SEAPLT.LIB(4)

This creates a SEAPLT library with four data blocks available for use.

To create the individual subroutine files in the SEAPLT library, the programmer must edit, or create, a

separate file in the library for each subroutine as indicated below:

```
EDIT SEAPLT/MAIN.FOR
INPUT
-----enter main program fortran code-----
EDIT SEAPLT/CONPRO.FOR
INPUT
-----enter CONPRO fortran code-----------
EDIT SEAPLT/TARGET.FOR
INPUT
-----enter TARGET fortran code-----------
```

or so on until all subroutine files have been entered.

B. COMPILING

Each file structure in the library file must be individually compiled in the form:

```
FORT library name/file name.OBJ,
LST=library name/filename.FOR
```

Compiler errors will be in the LST file, which can be read by typing the command

```
PRINT LST.LST
```

Because this becomes somewhat tedious for a file with several structures, one can create a batch routine which will compile the various subroutines. The following

coding will create a compile routine for a batch processor job.

```
EDIT COM.BAT
INPUT
FORT LIBRARY NAME/<1>.OBJ,
LST=LIBRARY NAME/<1>.FOR
EXIT
```

Using this routine, successive subroutines may be rather more expeditiously compiled by submitting the batch job

```
BATCH COM.filename
```

which will compile a single subroutine and must be repeated for each subroutine in the library.

C. LINK-EDITING

After a program is compiled, the program must be link-edited by typing the following commands:

```
LINK
OUT library name/RUN.OBJ
LINK library name/main program name
EDIT library name
EDIT SYS:DGL,SYS:RTL
```

Should the output of the link editor indicate that there are undefined terms in the program, this process must be repeated until all undefined terms have been resolved. This sometimes involves repeating the link-edit step four or five times.

D. EXECUTION

Executing the successfully compiled and linked program is, refreshingly, a very straightforward procedure. One merely enters the command

```
USR:  library name/RUN
```

If any execution errors are detected, they will be displayed on the screen along with any graphics output which has been generated before the error was detected.

LIST OF REFERENCES

1. Rogers, D.F. and Adams, J.A., Mathematical Elements for Computer Graphics, McGraw-Hill, 1976.

2. Sutherland, I.E., "Computer Displays," Scientific American, June 1970.

3. Sutherland, I.E., Srroull, R.F. and Schumacker, R.A., "A Characterization of Ten Hidden Surface Algorithms," Computing Systems, vol. 6, no. 1, March 1974.

4. Pennington, J.E., "Equations for CRT Display," NASA Langley Research Memorandum, June 16, 1971.

5. Burgen, G.E., Three-Dimensional Graphic Displays Design Specification and Analyst Manual (draft), Decision Science, April 1980.

6. Wise, R.S., "An Initial Examination of the Application of the Tektronix 4081 Graphics Computer to the Display of a Missile versus Aircraft Encounter," Masters Thesis, NPS, June 1980.

7. Tektronix, Inc., Tektronix Plot 80 DGSS Reference Manual, vols. 1 and 2, January 1979.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Professor R. E. Ball, Code 67Bp Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5.	Dr. George Burgen Decision Science, Inc. 4901 Morena Blvd. San Diego, California 92117	1
6.	LCDR Gray DeVeaux Hobby, USN Naval Plant Representative's Office Sperry Division Sperry Corporation Great Neck, New York 11020	1

(2nd card)

193820

Hobby

A user's manual and program description of SEA PLOT: a three-dimensional graphics program depicting a war-at-sea encounter between a single ship and a single aircraft.

Thesis
H6125
c.1

193820

Hobby

A user's manual and program description of SEA PLOT: a three-dimensional graphics program depicting a war-at-sea encounter between a single ship and a single aircraft.